

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Nazia Hussain

Requirements Engineering for Globally Distributed Teams using Scaled Agile Framework

Master's Thesis
Espoo, June 04, 2018

Supervisor: Professor Marjo Kauppinen, Aalto University
Advisor: Maarit Manninen, M.Sc

Aalto University
 School of Science
 Master's Programme in
 Computer, Communication and Information Sciences

ABSTRACT OF
 MASTER'S THESIS

Author: Nazia Hussain	
Title: Requirements Engineering for Globally Distributed Teams using Scaled Agile Framework	
Date: June 04, 2018	Pages: 86
Professorship: Software and Service Engineering	Code: SCI3043
Supervisor: Professor Marjo Kauppinen	
Instructor: Maarit Manninen, M.Sc	
<p>As large organizations are striving to deliver software at a faster pace and to keep up with the latest trends, they are in a transformation stage of adopting to Scaled Agile Framework (SAFe). SAFe is a framework for implementing agile practices at enterprise level and it provides a roadmap for portfolios, programs and teams. Large organizations adopting to SAFe are facing challenges in coordinating, planning and managing requirements, as they work with globally distributed teams.</p> <p>The goal of this thesis was to improve the Requirements Engineering (RE) process using Scaled Agile Framework in globally distributed teams. The main research method used in this thesis was action research, an iterative approach which combines theory and practice. The empirical study was conducted in a large project that used SAFe and had eight globally distributed teams. In order to investigate the challenges faced by globally distributed teams, analysis of the existing literature and RE process flow in SAFe was important. It served as a good input to understand which good RE practices can be applied in the empirical study.</p> <p>The results of the study show that visually representing requirements as models and sharing domain and system knowledge through Community of Practice (CoP) reduced ambiguity in requirements. The good RE practice applied in SAFe, of working and improving collaboratively with the globally distributed teams helped in better coordination and managing of requirements. In addition to this, it was also essential to have SAFe training to develop clear and shared understanding of the framework and RE process.</p> <p>The lessons learned from the empirical study indicate that a well-organized PI planning is the key RE practice of SAFe in providing the big picture of requirements to all members in distributed teams. In addition, Community of Practice (CoP) can be a key RE practice of SAFe in sharing knowledge such as business domain, system knowledge, skills and techniques, and experiences.</p>	
Keywords: Requirements Engineering (RE), RE process, Scaled Agile Framework (SAFe), globally distributed teams	
Language: English	

Acknowledgements

This thesis fulfills my purpose of coming to Finland and it is now time for me to thank everyone who has supported me in this journey.

I would like to thank my supervisor Marjo Kauppinen for always supporting and motivating me to do my best. Thank you for your valuable insights that helped me further with my study.

The other important support pillar for my work is my instructor Maarit Maaninen who was always ready to support and encourage on my research. I would also like to thank my employer and my co-workers for their part in making my writing as pleasant as possible.

I would like to express my sincere gratitude to my grandparents, parents and my brother for their continuous support and motivation in my life, without whom this journey would have been really difficult.

Last but not the least the people in my life- family, friends, mentors and professors from my programme who have been part of my development process. Thanks for all the smiles, laughter and happiness. You all have been very kind and supportive towards me.

Espoo, June 2018

Nazia Hussain

Contents

1. Introduction	6
1.1 Motivation	6
1.2 Research Problem and questions	7
1.3 Scope	8
1.4 Structure	8
2. Research Method	10
2.1 Literature review	10
2.2 Empirical Study	11
2.2.1 Case Description	11
2.2.2 Research Process	13
3. Literature study	17
3.1 Requirements Engineering (RE)	17
3.1.1 What is RE	17
3.1.2 RE Activities	18
3.1.3 Good RE Practices	20
3.2 Scaled Agile Framework (SAFe)	23
3.2.1 What is SAFe	23
3.2.2 Big Picture of SAFe model	26
3.3 Requirements Engineering in SAFe	29
3.3.1 RE process in SAFe	29
3.3.2 Epics, Features and Stories	32
3.3.3 Good RE Practices of SAFe	34
3.3.4 Program Increment (PI)	40
3.4 Summary	43
4. Empirical Study	47
4.1 Analysis of the case company project	47
4.1.1 Study of RE process using SAFe	48
4.1.2 Meeting Composition	56
4.2 Reflected Problems	57

4.3 Good RE Practices for Improvements	61
4.4 Lessons learned	68
5. Discussions	71
5.1 RQ1: Current RE process using SAFe in the case company	71
5.2 RQ2: Good RE practices applied in RE process using SAFe	74
5.3 RQ3: Lessons from applying good RE practices in globally distributed teams	76
5.4 Limitations of the study	78
6. Conclusions	79
References	82
Appendix	86

1. Introduction

1.1 Motivation

Agile methods have become a very popular approach for managing requirements and other software development processes. Their goal has been to enhance management and execution of software development projects by improving on-time delivery of projects, product quality and customer satisfaction [21]. According to [19] there has been an increasing number of companies adopting to agile as there is a 62% of acceleration in product delivery and 56% of ability to manage change requirements, hence helping the organizations around the world succeed. Originally the agile methods were designed for small organizations having single team projects [21]. But in recent years, they have become valuable for large organizations as well, despite the fact that they are difficult to implement [22], [29]. For example, agile methods has been adopted in many large-scale projects in organizations such as Nokia and Amazon. Scaling these methods has created challenges, such as synchronization and communication across large and globally distributed teams [11].

However, the momentum around scaling agile is growing rapidly and it faced a tremendous growth since 2014. Several methods and frameworks for scaling agile methods were created and according to [20], the most popular scaling agile method was Scaled Agile Framework (SAFe). It made a significant jump to become the most popular scaling agile method from 19% in 2014 to 28% in 2017 in comparison to other methods such as Scrum/Scrum of Scrums, Agile Portfolio Management (APM) and Large Scale Scrum (LESS) [20], as shown in table 1. During this transition, there has been changes in process and methods on managing requirements and other software development processes. For example, after a decade of identifying best requirements engineering (RE) agile practices which focused heavily on a team level, organizations started looking for scaled agile practices in order to scale the agile practices from team level to enterprise level. Table 1 shows a comparison of usage of different scaling agile methods and framework, based on the published report ‘State of Agile’ in 2016 [19] and 2017 [20] by Version One.

Table 1: Comparison of scaled agile methods and frameworks

Scalable methodology used (2016) [19]	Scalable methodology used (2017) [20]
72% Scrum/Scrum of Scrums	28% Scaled Agile Framework (SAFe)
27% Scaled Agile Framework (SAFe)	27% Scrum/Scrum of Scrums
23% Internally created methods	13% Internally created methods
17% Lean management	4% Lean management
9% Agile Portfolio Management (APM)	4% Agile Portfolio Management (APM)
6% Large-Scale Scrum (LESS)	3% Large-Scale Scrum (LESS)

In this thesis, the goal is to examine the requirements engineering process in large organizations using Scaled Agile Framework (SAFe). We analyze the requirements engineering activities at each level in SAFe and identify good RE practices to improve the RE process in SAFe. The research study is done by applying action research method to the case company project where I will be working as a business analyst for the case company project. This research study ensures that there is common understanding of requirements across the globally distributed teams using Scaled Agile Framework.

1.2 Research Problem and questions

As organizations are striving to deliver software at a faster pace to meet their dynamic customer needs and to keep up with the latest trends, the globally enterprise IT organizations are in a transformation stage of adopting to Scaled Agile Framework (SAFe) to execute IT projects. SAFe is a framework for implementing agile practices at enterprise level and it provides a roadmap for teams, programs and portfolios to scale their agile teams. It is noted that there are many challenges and improvements required to adopt SAFe especially in the requirement engineering phase, where large projects require good coordination and communication across teams, and manage dependencies [24]. Scaled Agile Framework is a different approach from a traditional IT hierarchical organizational framework which also leads to a significant cultural change that affects the overall requirement engineering process.

This study sets out to find out through action research in refining the execution of the requirements engineering process using Scaled Agile Framework. It is done with a case project in a multinational IT company which is among the leading independent information technology and business process services firms in the world. The study aims to answer the below research problem and questions.

Research Problem for this study is:

- *How can the Requirements Engineering (RE) process using Scaled Agile Framework be improved in globally distributed teams?*

The problem will be further studied with these three research questions:

- **RQ1:** *What is the current state of Requirements Engineering (RE) process using Scaled Agile Framework in the case company?*
- **RQ2:** *Which good RE practices can be applied in the RE process using Scaled Agile Framework, of the case company?*
- **RQ3:** *What lessons can be learned from applying, good RE practices in globally distributed teams using Scaled Agile Framework?*

1.3 Scope

This thesis is based on my own and my team member's experiences in a large organization which has adopted to Scaled Agile Framework (SAFe). It is centered on improving the case company's RE process to achieve a common understanding of requirements and customer needs in large teams.

The research on primary studies attempts to provide a deeper knowledge on requirements engineering activities and good practices. It later provides an insight to SAFe and then summarizes the RE process in SAFe. Furthermore, the thesis extends the pool of primary studies about requirements engineering in SAFe by conducting an action research in the case company. It focuses on overcoming the challenges faced in case company project, by applying good RE practices during the RE process. The good RE practices are based on keeping in mind, the lean-agile mindset and its core values.

The good RE practices should be possible to take into use in a relatively short time-span. The scaled iteration also known as program increment in SAFe has 8-12 weeks of duration of delivering requirements as features. This enables my research study to review and validate the outcomes of the solution proposed in the case company project.

1.4 Structure

This thesis starts with the introduction chapter, where the research problem and research questions are mentioned.

Chapter 2 summarizes how the research in this thesis was conducted. First, the literature review and the approach to that is discussed. The following section describes the action research method used for empirical study, the research method used for this study, and how it was followed throughout the course of this work

Chapter 3 is the Literature study which is divided into four main sections. The first section provides an overview of requirements engineering activities and practices followed in agile development projects; the second section Scaled Agile Framework (SAFe) covers the information about scaling agile into large enterprise organization. The third section gives a holistic explanation of how requirements engineering takes place in SAFe. Finally, a summary of analysis is done on the RE process of SAFe and the good RE practices that can be applied in SAFe.

Chapter 4 is the Empirical study which uses the action research method. First it analyzes the current situation in the project with reference to the literature; then identify the problems and propose good practices to improve the RE process in SAFe. The execution of the suggested

practices and reflection of the results is also discussed in this chapter. Empirical study concludes by summarizing the lesson learnt from overall execution of good practices and interaction with the case company project.

Chapter 5 discusses on how the research questions were answered during the course of this thesis. This includes a summary of what are the lessons learned from applying good practices in RE process using Scaled Agile Framework. The limitations for this thesis is also discussed in this chapter.

Chapter 6 is the conclusion. It inspects how successfully the research problem was answered and provides insight for future research.

Figure 1 shows how the research questions relate to the contents of this thesis. It focuses only on showing how the research questions relate to the literature study and the empirical study.

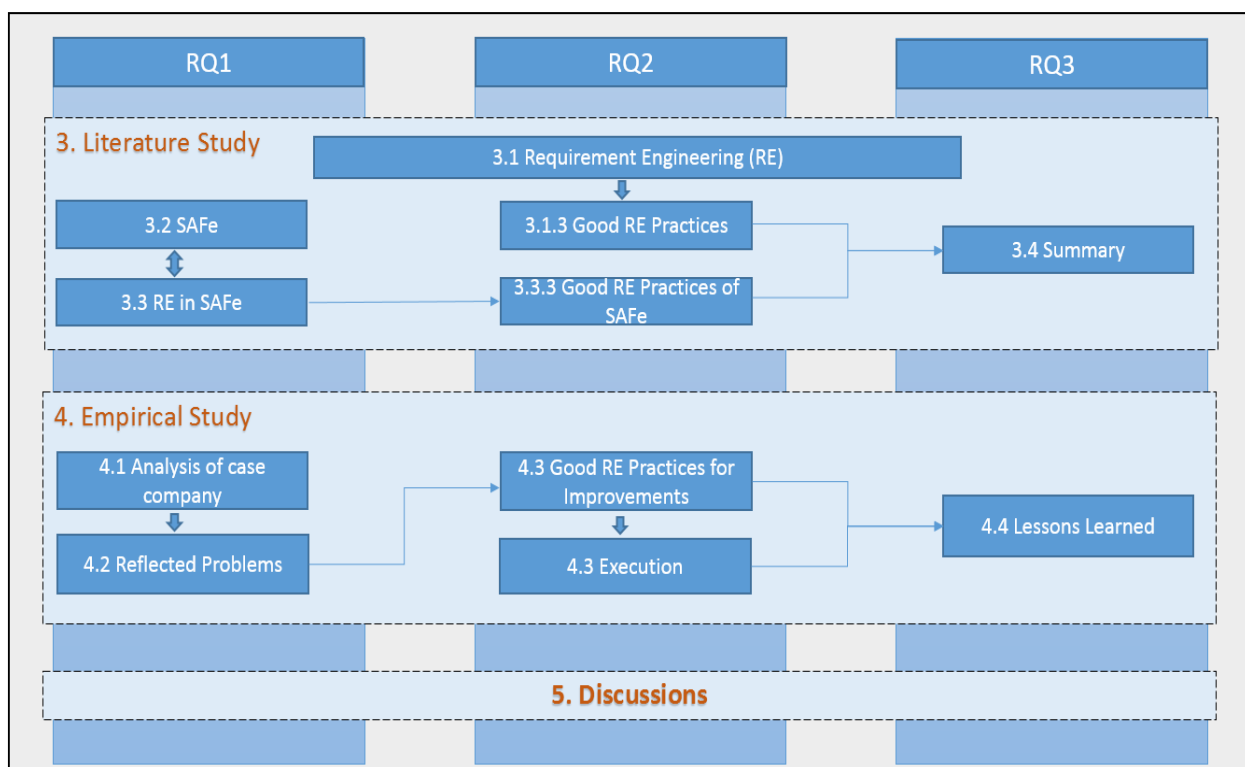


Figure 1: Relationship between research questions and structure of the thesis

2. Research Method

This chapter summarizes how the research in this thesis was conducted. First, the literature review and the approach to that is discussed. The following section describes how the empirical study of this thesis was done.

2.1 Literature review

The literature review in this study looks at three research questions: *RE process in Scaled Agile Framework (SAFe)*, *what good RE practices can be (or were) applied to improve the RE process in SAFe* and *what lessons can be learned from applying the good RE practices in globally distributed teams using SAFe*. The research was mainly based on the academic articles, books and SAFe official site. There was a limitation in finding articles on SAFe as it is relatively a new framework. It was challenging to find research articles or scientific publications specific to requirements engineering in SAFe. Hence the SAFe books and their official site were used to support my literature study. In addition to this, SAFe training and certification was taken to get a deep understanding of this framework.

The information was analyzed by establishing an extensive literature review by doing a comparative analysis for the research questions. The papers used in this research were collected from the Aalto university database, articles from the course materials and google scholar. Below is the list of sites and databases used to access academic articles and publications.

- *Nelli Portal (Aalto Library)*: <http://www.nelliportaali.fi>
- *ACM Digital library*: <http://portal.acm.org/>
- *IEEEExplore*: <http://ieeexplore.ieee.org>
- *ScienceDirect*: <http://www.sciencedirect.com/>
- *Google Scholar*: <http://scholar.google.com/>

The common keywords or the search strings used, is shown in the table 2. They are divided into two categories. The queried results were huge for requirements engineering in agile software development but were limited for SAFe. Hence a combination of keywords such as, ‘requirements’, ‘scaled agile’, and ‘large organizations’ were used. The search resulted in articles related to the category of ‘enterprises adopting to scaled agile methods’. However when the key word ‘Scaled Agile Framework’ was used, it resulted in very few articles, highlighting the newness of the topic in the academic world.

Table 2: Categories and keywords of the literature review

Category	Keywords
Requirements engineering	Requirements AND Agile, good requirements engineering practices, requirements AND large organizations
Scaled Agile Framework	“Scaled Agile Framework” , scaling agile AND large organizations, safe AND agile, scaled agile practices

For the first research question, the search was divided into two steps, the initial step was to understand what is requirements engineering in agile and then analyze the RE process in SAFe. The data in the literature review is provided with a comparative analysis with different authors. For the next step of analyzing the RE process in SAFe, the information was mainly collected from SAFe book and website.

For the other two research questions, a partial systematic mapping process was applied. The systematic mapping process includes steps, such as, definition of the research questions, search for primary studies, screening and analyzing papers, key wording of abstracts and data extraction [47]. The first step of this process was the identification of research questions. The goal was to identify the good RE practices that can be used to improve the RE process using SAFe. Initially the research was started by identifying the good RE practices for agile software development. It was done by applying a systematic mapping process using different articles published by authors who did a case study on agile development organizations. In the next step we analyzed if these practices can be applied in SAFe, along with the other SAFe RE practices.

There are not many articles, where research has been done on good RE practices of SAFe in large organizations, or of their related challenges and benefits. Hence, the good RE practices of SAFe were identified and analyzed from SAFe books by applying extensive literature review. However it was not feasible to gather adequate and complete knowledge through review of literature alone. In this research, literature can only be used as a theoretical starting point for identification of the practices.

2.2 Empirical Study

This chapter introduces the case company and the target project for the empirical study. The research process used is action research which is explained in detail. In addition to this, it also explains how the data was collected and analyzed for the empirical study.

2.2.1 Case Description

The **case company** for this research is a global software service company, supporting a financial insurance based IT project. It is located in Helsinki Finland and currently has more than 500 employees working in globally distributed teams.

It offers software and services for insurance and administrative information systems outsourced by a well-established bank group. The company implements an IT service model using Scaled Agile Framework (SAFe) which works closely with the customer and at the same time have the opportunity to work with other branches of the company located in different locations such as India and Poland.

The company provides services in two divisions, ‘application maintenance’ and ‘application development’. With ‘application maintenance’, the customer’s IT applications are in a process of constant improvement with respect to performance, stability and usability. As and when the regulatory requirements or business needs change, there is a small development of application maintenance. ‘Application development’ projects are implemented based on the business requirements and are developed in accordance to Scaled Agile Framework. The company is also a Gold Scaled Agile Partner and provides services for the employees with training and coaching in SAFE framework.

The **case company project** for this research study comes under the company category of application development and is implemented using Scaled Agile Framework. This project is termed as ‘agile release train (ART)’ in SAFe language and has globally distributed teams consisting of 100 employees working in both onshore (Finland) and offshore (India and Poland) locations.

Case company project named ‘Project X’ is a SAFe Release train of one program level having 8 agile teams and 100 members both onshore and offshore. The agile teams are organized based on domains and technology environments. Figure 2 explains the train and team structure of the case company project. As part of my research study, I will closely be working with team 4 as a business analyst from Finland location and will also be interacting with members from the other teams.

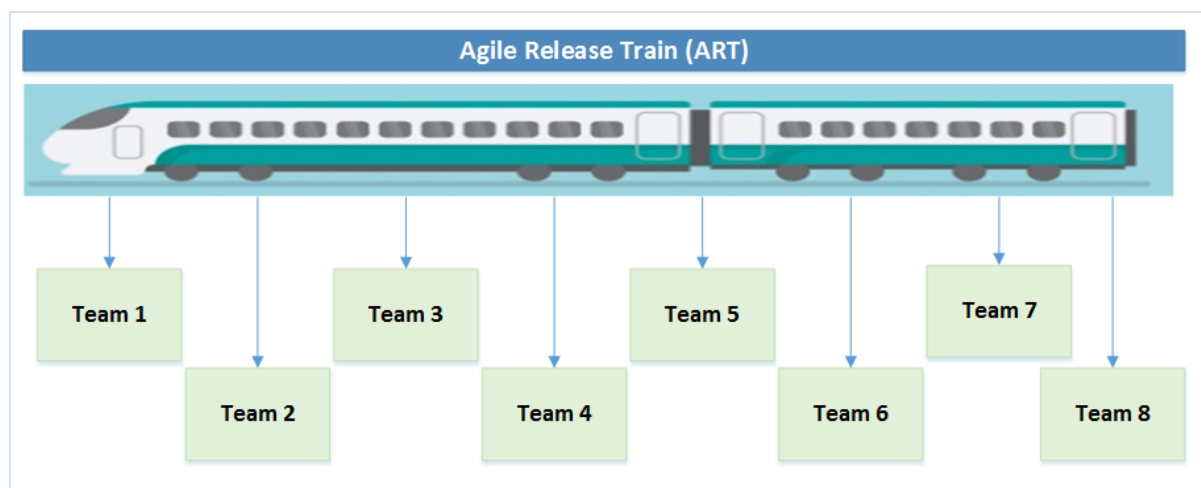


Figure 2: Structure of the case company project

Based on my interaction with the members working in the company projects using SAFe, faced challenges in establishing the core foundation of common understanding of requirements in large distributed teams. Since the issues or challenges were common across the organization, any project within the organization using SAFe framework qualifies for my action research method. This project was selected among other projects using SAFe, due to its large size of the teams distributed globally, hence making it an ideal case to answer my research questions.

The research focuses on understanding the RE process in case company project using SAFe. But the underlying goal is to establish good RE practices that enables a common understanding of requirements across globally distributed teams. Therefore, this case company is a good platform to perform my research study.

2.2.2 Research Process

The goal of this thesis is to improve the RE process to deliver a common understanding of requirements using Scaled Agile Framework. We focused on three areas, first we wanted to understand the RE process in SAFe. Second, we wanted to identify some good RE practices that can be applied to improve the RE process and lastly, what lessons can be learned by applying these good RE practices in globally distributed teams, using SAFe.

The case company project is using SAFe framework and is in a process of improving and adapting to changes. This study sets out to find out through action research in refining the execution of the RE process in SAFe by applying good practices.

Action research method is a ‘combination of theory and practice’ performed in an iterative process [46]. It recommends to involve researchers and practitioners to participate together to solve the real world problems [46]. In action research, more than what practitioners says they do, it emphasizes on what they really perform. It inspires the researcher to intervene, experiment and finally learn from the process through an organic iterative process. The method includes activities such as, problem diagnosis, action intervention and reflective learning [46]. However, according to [38] it can also be divided into five phases that can be iterated. The phases include, problem diagnosis, action planning, action taking, evaluating and specifying learning. In my case, the action taken and evaluation was performed simultaneously as it went hand in hand together. Figure 3 gives an overview of the research process used in the empirical study.



Figure 3: Action research process

Problem diagnosis: The first phase of this study was to do a current state analysis of the case company project focusing on requirements engineering (RE). This enabled to develop theoretical understanding about the nature of the organization and identify the problem areas. It was performed by conducting interviews, making observations as a business analyst in an agile team, taking feedback from the team members and getting to know the customer.

Planning action: After analyzing and identifying the reflected problems, we proceeded with planning and identifying some good RE practices that can help to solve the problems and improve the RE process in SAFe. It was also important to plan the scope and schedule to implement these practices for the program increment (scaled iteration). In the first draft, it was planned to execute the practices for one iteration, which later got extended to the next iteration. The plan was represented in a document of how the empirical study was going to be done and what practices can be applied. The plan was validated by the SAFe coach who is also an advisor for my research study.

Action taking and evaluation: In this phase, the real action was taking place which is, implementing the suggested RE practices at the team and program level. The practices were executed for two iterations (2 program increments) and each iteration took 8-10 weeks. During this process, I was actively participating in a team as a business analyst from the client location and contributing to the process of requirements engineering in SAFe. Simultaneously, the results were observed and evaluated during the process. The results were collected from retrospective meetings, interviews, feedbacks and by having open discussions with the team members and clients.

Reflective learning: After the suggested good RE practices were tried and applied in the case company project to improve the RE process in SAFe, it was important to understand what were the lessons learned during the process. At the end, a survey was taken to identify what were the good RE practices that solved most of the challenges to all team members of different roles working in different locations.

The main reason to apply action research method is that, it inspires the researcher to intervene, experiment and learn from the process iteratively. This process helps to analyze different requirement engineering activities and practices followed in the case company and apply different requirement engineering practices that can solve the problem description. The key here is the reflective learning step, which can change perspective on the research topic and add value with each iteration. The main purpose of this process is to gain knowledge whether it is a success or a failure.

2.2.3 Data collection and analysis

Data was collected continuously by using several methods such as interviewing, participating in group discussions, interacting with team members, taking feedbacks and by making observations. Also, my personal experiences in the case project represented a big part of the research data.

The research questions were addressed by conducting interviews with major stakeholders and members such as Release Train Engineers (RTE), scrum masters, product owners, developers and testers. The interview questions were created based on the role of the member. For example the interview questions for RTE were different from the questions created for a developer. In total, there were 13 interviews taken, where two interviews were conducted for two different RTEs, two for scrum masters, three for product owners, five for developers and testers, and one with the SAFe coach. The category and list of interview questions are mentioned in the Appendix section of this thesis. In addition, my involvement with the project as a business analyst helped to gather more realistic and quantitative information. Added advantage was to interview a SAFe coach who already had a good experience in facing and solving issues across projects in the organization.

Over the course of this process, a **research diary** was created to gather all the information and analyze my observations. A research diary, helped to analyze both successful and unsuccessful routes of learning. It prompts insights which informs a variety of methodological and theoretical decisions in relation to the research [10]. From different data collection methods, a lot of raw data was collected in the form of notes in the research diary. I then analyzed by carrying out the following steps:

- **Grouping information based on user roles:** When the data was collected through interviews or interaction with the team members. I started to put them into the research diary by making sections based on their roles. For example, the data from the developers were collected under one section and the data from the product owners were collected in other section.
- **Compare and prioritize responses:** The information was again compared from the responses coming under the same section of user role and prioritized based on the research questions. For example, the responses from different product owners were compared and filtered based on the most common issues, suggestions and feedback.
- **Create a mind map:** A mind map is a diagram to visualize the information, since there was a lot of raw data to categorize, compare and prioritize, the mind map concept was very helpful to connect and visualize the information. The mind map was used both in the above two steps.

Hence, the different data collection methods, research diary and creating mind map together created a very valuable tool to collect information related to my research questions.

3. Literature study

This chapter is divided into three sections, the first section 3.1 describes the RE process; the second section 3.2 gives an overview about the Scaled Agile Framework and the third section 3.3 describes the RE process in a Scaled Agile Framework.

3.1 Requirements Engineering (RE)

In order to substantiate my research study on how to improve RE process in SAE. It is first important to understand what is RE process and its good practices. In this chapter, I am making an attempt to provide the relevant literature study analysis on agile requirement engineering process that includes activities such as requirement elicitation, analysis, representation, validation and requirements management. Further, in this chapter good agile RE practices are explained that helps in improving the RE process.

3.1.1 What is RE

There are many and different ways of defining what is requirements engineering, but before we get into that, it is first important to understand the meaning of the term ‘requirement’. A requirement is a necessary and a key attribute in a system. It can be a statement that identifies a capability, characteristic or quality factor of a system in order to have value and utility to a customer or user [1]. According to Sommerville, software requirements refer to the description of the purpose that a system is intended to, they can be regarded “as a specification of what should be implemented” [13]. In general, requirement is a need that what a customer really wants build to obtain a successful software system. It is very important to gather correct set of requirements by understanding the user needs so that the software system is implemented correctly by the technical team.

Requirements engineering (RE) is a term, which is used to describe the process of creating requirements for a system [1]. Success of a software system is determined by how effectively it meets the customer expectation. In order to meet customer expectations, it is very important to establish a requirement engineering process which drives the software system evolution. Today there are many different definitions of requirements engineering and in my opinion the most precise definitions are given below:

According to Nuseibeh & Easterbrook [2]:

‘Requirements engineering (RE) is a process of measuring the success of the software system to the degree which it meets the intended purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation’.

And according to Zave [14]:

‘Requirements engineering is defined as the branch of Software Engineering concerned with the real-world goals for, functions of, and constraints on software systems; it is also concerned with the relationship of these factors to precise specifications of software behavior and to their evolution over time and across software families’.

From the above definitions, it can be analyzed that requirements engineering is one of the initial steps in software project model where the discussion starts on what needs to be built and details the ‘what’ and ‘why’ of the system to be built. It denotes what the customer is really looking for, the functionalities that the system should deliver to satisfy the customer and also denotes the constraints of the system. This is a critical process that needs to be well defined, understood and maintained in any software development organization. Overall, requirements engineering contributes the success of the software system, the cost effectiveness, timely delivery and customer goals.

3.1.2 RE Activities

Requirements engineering (RE) consists of systematic and repeatable activities that ensure the completeness, consistency and relevance of the system requirements [3]. Figure 4 illustrates the typical RE activities: elicitation, analysis, representation, validation and requirement management.

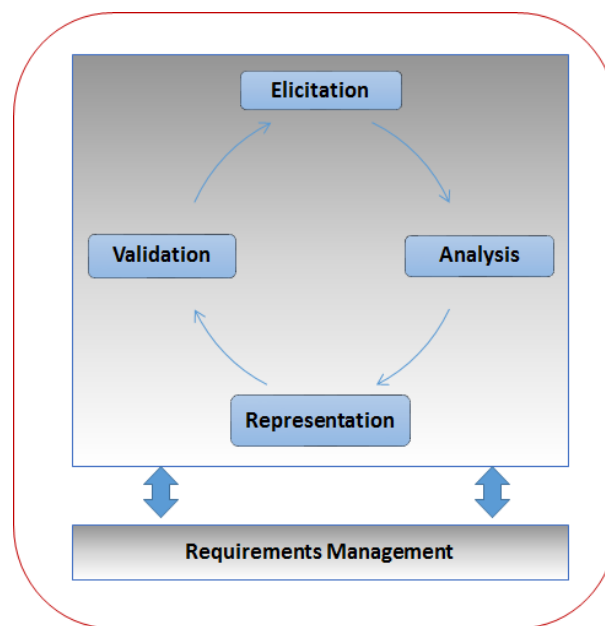


Figure 4: Typical RE activities

Requirement elicitation

Requirement elicitation is the first step in the RE process. It is the activity of discovering customer and user needs. According to [2], the term "elicitation" means "to capture".

This activity includes discovering the requirements (the necessary and the hidden requirements) and understanding customer needs for the system to be built. After performing the groundwork of identifying main stakeholders, during this process a continuous effort will be made to conduct joint application development discussions, interviews and prototype presentation to collect the goals of the system to be built [17]. There are many different techniques that can be used for elicitation process such as, Questionnaires/Surveys, Group Discussion, Scenario-based Discussions, Whiteboard Sessions/Interviews, Prototyping and Goal-based Discussions [17].

Requirements analysis

In this activity the initial set of user needs and requirements are refined and prioritized. It helps in identifying essential needs from the perspective of users and also in identifying conflicts and inconsistencies [13]. As Analysts, certain questions could be asked to the customer during this analysis phase, which will help them to identify and prioritize the requirement. There are mainly 5 different questions that can be asked as part of the analysis [18]: What is the purpose (goals)? What objects are involved? Where is the system located? When should things happen? Why is the system necessary? In my opinion, these questions can help in analyzing if the requirement is necessary, consistent and feasible in the context of the planning, budget and schedule for the system development. It can also help in identifying architectural impacts and dependencies.

Requirement representation

This activity includes representing or modelling the user needs and constraints collected as a result through elicitation and analysis activities. According to Lauesen [5], a good requirements specification should fulfill eight criteria of quality. The eight criteria's are, the requirement should be correct, complete, unambiguous, consistent, ranked for importance and stability, modifiable, verifiable and traceable. There are several practices and methods to represent requirements such as user stories, use cases, prototypes or wireframes, videos, conceptual diagrams and domain model [50]. In my opinion, representation of the requirement in the form of prototype or modelling gives a good visual understanding to the users along with use case or a user story.

Requirements validation

The purpose of this phase is to ensure that requirements are complete, consistent and clear to satisfy all stakeholders [3]. It is important to communicate the requirements with relevant stakeholders and have them validated before they are implemented. During this review process the findings can be collected from stakeholders and those findings or review comments can be accommodated for the completeness of the requirement documentation. Once the review process is completed, all requirements are formally approved by stakeholders. If there is an exception or conflicts in requirements, they need to be reworked and agreed for future release purpose or as a change request [3].

Requirements Management

This is an overall process of managing the activities of scheduling, coordinating, and documenting the requirements engineering activities. According to Sommerville [7] requirements management is the process of understanding and controlling changes to system requirements. It supports an established process when there are requirement change requests and then link these change proposals to original system requirements. It also supports in establishing a link between dependent requirements so that the project team can assess the impact of requirement changes. The link here means, tracking of the requirements which is an important activity under requirement management, the term used is traceability.

Requirement traceability is a major part of the requirement management process, which determines how easy it is to read, navigate and change requirements documentation [2]. Traceability is established between different components like functional requirements, development components and test cases/results in both forward and backward direction [8]. Similarly another major activity under requirement management is change management. **Change management** refers to the ability to manage changes to requirements throughout software development lifecycle [8]. This process also addresses any change requests during the evolution of the software. Any changes or gap in the requirements drives a change management process. An effective requirement management process and tools helps to achieve the business goals effectively [8].

3.1.3 Good RE Practices

In this section, good RE practices are selected from the industry observations and results, which could also be used in the scaled agile projects. There are many studies focusing on the agile requirement engineering practices adopted by organizations. Hoffman and Lehner [6], identify some traditional good RE practices followed by successful RE teams. The practices are identified by focusing on areas of knowledge, resources and process. They have also been analyzed against cost of introduction, cost of application and key benefits.

Also, an empirical study on agile RE practices was performed by Ramesh and Cao [9]. They analyzed and identified seven agile RE practices by collecting data from 16 organizations that use agile methods. In addition to this, it was also important to identify what good RE practices can be used to overcome RE challenges for globally distributed teams. According to a study performed by Bhat, Gupta and Murthy [16] they analyzed the real-life case studies of distributed teams to come up with best RE practices. The practices are categorized based on people, process and technology by applying success factors such as shared goals, shared process, trust, shared culture and shared responsibilities. The table 3 lists some good RE practices for agile software development teams.

Table 3: Good RE practices

Good RE practices	References
<i>Face to Face Communication</i>	[3], [9], [16]
<i>Prioritize requirements</i>	[3], [6], [9], [12], [16]
<i>Modelling and Prototyping</i>	[2], [3], [6], 9, [16]
<i>Use review meetings and acceptance tests to validate and verify requirements</i>	[6], [9], [12]
<i>Test driven development</i>	[9], [15]
<i>Iterative Requirements Engineering</i>	[9], [16]
<i>Train team members to use right processes, tools and technology</i>	[4], [6], [16]

The good RE practices are described further in detail. These practices can be used as potential good RE practices in case project using SAFe from the perspective of empirical study.

Face-to-face communication

This practice helps in understanding the real needs of the customer and reduces ambiguity across team members [9]. Though organizations create formal documentation, frequent customer interaction is needed to alleviate the ambiguity in documentation and to work on requirements that is not outdated from its time of origin. Also, direct interaction always helps to improve the trust between customers and team members. For the globally distributed teams, it is recommended to get the teams together at the formation stage for a face-to-face kickoff session [16]. Questioning and interviewing are other good ways of engaging customers for a good quality software product, it also helps in discovering the hidden requirements [3].

Prioritize Requirements

Agile software development starts with focusing on high-priority requirements so that customers get the maximum business value [9]. Customers are responsible for prioritizing the requirements that provides them the greatest benefit. The technical team is welcome to provide their inputs such as technical risks, cost and difficulties. Based on the collective inputs, customer can change the priority of the requirement [3]. From the reference section in table 3, it can be observed that, there have been many studies who have recommended this practice for a successful RE process. Below steps can be considered for requirement prioritization [12].

- *Development team*: Estimation of implementation time and risk factors
- *Customer*: Business Priority
- *Customer & Development Team*: Finalize Prioritized Feature for Implementation

Modelling and Prototyping

Prototype of a system helps in eliciting and validating system requirements [3]. It is a visual and initial version of a system or functionality which is available before the development process. According to Nuseibeh and Easterbrook [2] prototyping has been used to resolve ambiguity in critical projects where early feedback from stakeholders is provided. It can also be used to provoke discussions with customers to enable them to think on some missing requirements. This process eliminates the time consumed and overhead of creating formal requirement documents.

Hoffmann and Lehner suggests to include complementary models together with prototyping to eliminate specification ambiguities and inconsistencies [6]. The different categories of modelling are enterprise or system modelling which is used to capture the purpose of the system; data modelling for representing information systems; behavioral modelling for representing functional behavior of stakeholders and system; and domain modelling for providing abstract description about the domain [2]. From the reference section in table 3, it can be observed that, there have been many studies who have recommended this practice along with requirements prioritization to be applied for a successful RE process.

Use review meetings and acceptance tests

This practice helps to validate a common understanding of requirements between developers, testers and customers. Organizations schedule frequent review meetings to validate the requirements [9]. The review meetings are helpful to assess the project progress, to increase the trust between customer and development team and to identify problems early during development cycle [12].

Acceptance tests are the other way of validating and verifying requirements through QA personnel. Several organizations find implementing such testing difficult owing to the difficulty of access to the customers who develop these tests [9]. Hence, many organizations use QA personnel to help customers develop these tests.

Test-driven development

Mostly referred as TDD, this is an approach or practice where developers creates tests before writing a new functional code which specifies the system's behavior [15]. These test scripts capture requirements and design related to the product delivery. Test driven development improves the efficiency of the tests and code quality for each iteration as higher level of traceability is available for developers [15].

Ramesh and Cao [9] conducted a study based on the data collected from 16 organizations that implemented agile practices. According to this study, Test-driven development (TDD) is the least adopted practice in comparison to the other practices, as only 6 organizations adopted to it. The reason being that developers are not accustomed to a discipline structure [9].

Iterative Requirements Engineering

The main concept of iterative engineering is well connected to the customer comment that is, “I’ll know it when I see it” [9]. This helps the development process to start with a good level understanding of requirements. Agile teams work in iterations for 2 weeks known as sprint cycle. In the beginning of each sprint, requirements are discussed in detail with development and testing team and are constantly validated with customer. Customer iteratively improves the requirements path by continuously monitoring and experiencing the progress through iteration demos. Iterative requirements engineering helps in establishing trust and good relationship with customers by delivering the output in iterations [9].

Train team members to use right processes, tools and technology

The teams without adequate training and coaching struggled with applying agile practices correctly in the RE process [4]. One of the practices used to improve the RE process was training the members to understand the process, tool and technology [16]. The practice of training members was also considered as one of the success factors that supported organizations for wide implementation of RE processes [4]. The purpose of the basic training was to describe why RE is important, to give an overview of the RE process, and to show how this process relates to the organization’s product development process.

3.2 Scaled Agile Framework (SAFe)

This section gives an introduction to Scaled Agile Framework, followed by its core principles and values. It further explains on how this framework is divided into different levels to manage the agile methods. The content and information about SAFe for literature study was mainly taken from the official website of SAFe [25] and the book [26].

3.2.1 What is SAFe

SAFe is a framework for implementing agile practices at enterprise level by providing a roadmap for portfolios, programs and teams [26]. Developed and co-founded by Dean Leffingwell, the Scaled Agile Framework is a structured template that supports large-scale organizations to embrace agility across the whole enterprise [26]. It is based on Lean-Agile principles [26] and incorporates the values of agile, as outlined in the Agile Manifesto [30]

- *Individuals and interactions over Processes and tools.*
- *Working software over Comprehensive documentation.*
- *Customer collaboration over Contract negotiation.*
- *Responding to change over following a plan.*

SAFe synchronizes alignment, collaboration and delivery for multiple agile teams [25]. It supports small scale solutions employing 50-125 practitioners, as well as, complex systems that require more than thousands of employees [25]. Case studies on the SAFe website show [31] that many large organizations have adapted this framework well and are getting outstanding business benefits from applying SAFe over the past few years. SAFe offers organizations and participants the possibility to increase competitiveness, productivity and quality [25].

Principles and core values

The SAFe core values and principles helps in defining the framework. The core values represent fundamental beliefs of the organization. SAFe comprises of four core values [34]:

1. Alignment
2. Built-in Quality
3. Transparency
4. Program Execution

SAFe also defines its principles [26] [34] which are considered to be the fundamental and basic foundation of this framework. It encourages to bring more productivity, solution quality, time to market and employee engagement. The founders of SAFe agree that it is always a challenge in applying agile methods to large enterprises and there is no off-the-shelf solution to the unique challenges that every enterprise faces [26]. These principles therefore contribute as a guidance to the enterprises to customize and apply the SAFe RE practices appropriately. However, the success of this scaling framework mainly depends on each enterprise and its culture to adapt the change.

These principles also highlight the core values of SAFe that drives the practices to scale agile for enterprise organizations. SAFe claims that these principles have evolved from agile principles and methods, lean product development, systems thinking, and observation of successful enterprises [26]. Table 4, summarizes the nine SAFe principles with description [26] [34].

Table 4: Principles of SAFe

Principles	Description
Take an economic view	In order to achieve best value and quality at shorter lead times, it is essential to develop a value stream economic framework which illustrates the costs of delay, operational and development costs [26] [34].
Apply systems thinking	It is essential to apply systems thinking in the scaled agile environment, which mainly focuses in understanding the system by investigating the interactions among the components that make up the system [26].
Assume variability; Preserve options	Responding to emerging requirements. Maintain multiple requirements and design options for a longer period in the development cycle [26] [34].
Build incrementally with fast, integrated learning cycles	Promoting to develop and build solutions incrementally in a series of short iterations [34].
Base milestones on objective evaluation of working systems	Each integration should provide an opportunity to evaluate the solution, frequently and throughout the life cycle of the project. This in turn helps in financial, technical and fitness for purpose [34].
Visualize and limit WIP, reduce batch sizes, and manage queue lengths	This principle suggests to limit work in progress (WIP) items and make it visible to all stakeholders. It recommends to consider only achievable and smaller chunk of requirements that can be achieved with lesser wait time [26] [34].
Apply cadence, synchronize with cross domain planning	Cadence transforms unpredictable events to predictable and provides a rhythm for development [34]. Synchronization causes multiple perspectives to be understood, resolved, and integrated at the same time [34].
Unlock the intrinsic motivation of knowledge workers	This principle emphasize on the limit of role of compensation and other factors to consider like, mutual influence within the team, minimal possible constraints, mission and autonomy with a purpose [26].
Decentralized decision making	Achieving fast value delivery requires fast decentralized decision making across different SAFe levels: portfolio, program and team [26] [34].

3.2.2 Big Picture of SAFe model

The “Big Picture” of SAFe model as shown in the figure 5 represents the holistic visual representation of the framework, highlighting the levels and important roles [25] [26]. The architecture of SAFe consists of three basic levels: *Portfolio level*, *Program level* and *Team level*.

The *portfolio level* is the highest level in the framework and guides the organization with their vision [26]. It provides high level business requirements which are represented as epics. The *program level* is responsible to implement the epics. It is done by breaking them into features, managing dependencies across agile teams, validating the solution and approving the deliverables during PI Planning [26]. The *team level* consists of the agile teams consisting of a small group of dedicated individuals, who are responsible to define the features into user stories, implement and test them in a short time box. The Agile Release Train (ART) in the program level, is a SAFe term for managing multiple agile teams having the same target [25]. Further in this section, we will be focusing on each level of SAFe and describe the important roles related to RE process.

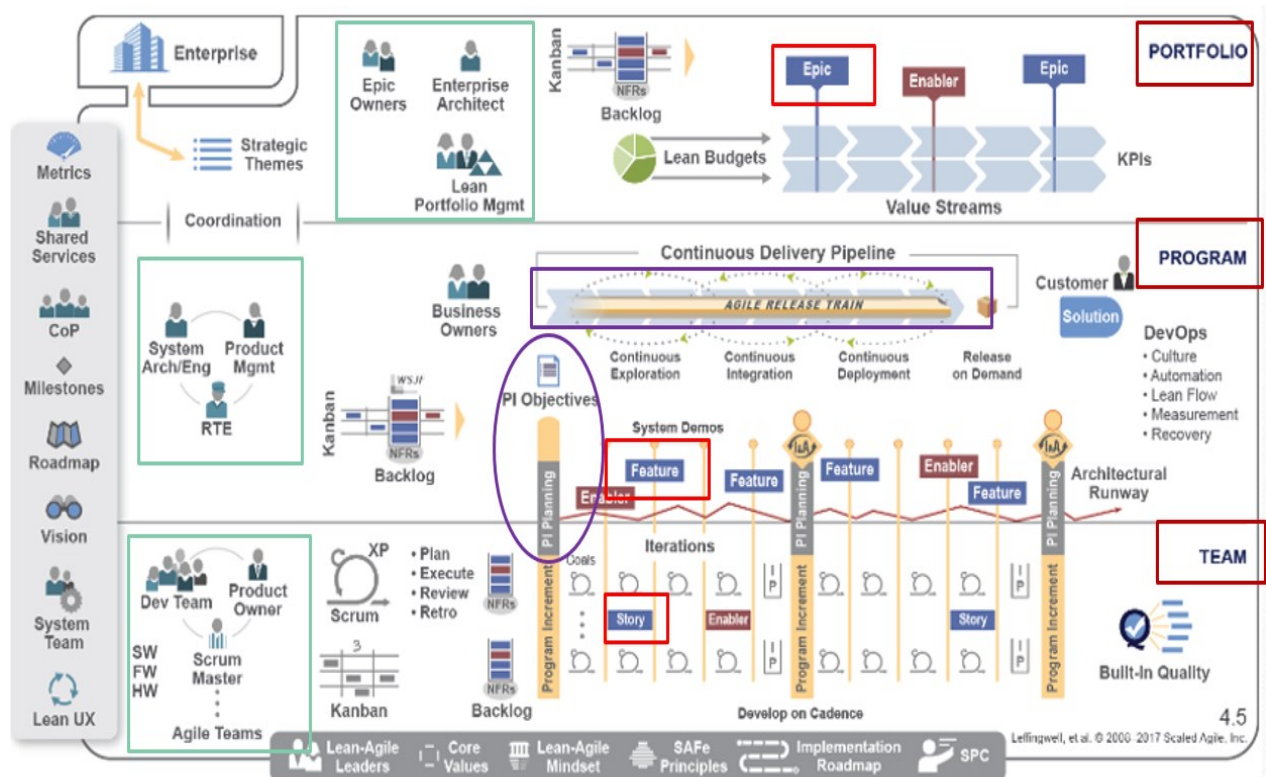


Figure 5: Big Picture of SAFe model [25] [26]

Portfolio level

Portfolio level is the highest level in SAE. It provides basis for guiding the enterprise or organization in their mission, and allocate funds and governance mechanisms to ensure that their strategic objectives are met [25]. The large enterprises could have multiple portfolio levels, whereas, the small and mid-size companies could have only one portfolio level [26].

It consists of lean portfolio management system that manages the portfolio level activities such as managing business, investments and financial constraints across ARTs [25]. They also contribute in providing strategic themes which defines the strategy of an enterprise. The epic owners and enterprise architects helps in contributing to high level business requirements which are represented as epics. Epics are filtered according to the strategic themes and stored in the portfolio backlog. The epics in the portfolio backlog acts as the highest priority in the framework which will be the input to the ART in the program level [25].

Program level

The next level of the framework is program level. It is responsible to implement the epics by breaking them into features and approving the deliverables during PI Planning. PI planning is a face-to face event which includes stakeholders from all levels [26]. The program level being the most important part of the SAE in an organization, consists of several Agile Release Trains (ART) [25].

Agile Release Train (ART) is created at the program level in SAE (as shown in figure 5). It is a long-lived, self-organizing team of agile teams. SAE recommends to have 5 to 12 agile teams organized in one ART, which includes totally 50 to 125 individuals [25]. The agile teams in ART is a self-organizing, self-managing, cross-functional group of individuals that delivers valuable, tested, working system every two weeks [26].

The Agile teams within ART consists of five to ten people and are aligned to a common mission to achieve common business and technology goals [25]. It uses a team framework which combines the best of Scrum practices such as sprint planning, sprint review, sprint retrospective and daily stand ups [29]. In addition to these events, SAE introduces the ‘release planning’ to synchronize teams for deploying incrementally after every iteration [33]. Table 5 includes the dedicated roles and their responsibilities at the program level related to RE process.

Table 5: Roles and Responsibilities at Program level [25] [26]

Release Train Engineer (RTE):	Acts as the chief scrum master for the whole Agile Release Train. His responsibility is to facilitate the program level process and execution, manages risks and dependencies.
System Architect-Engineering:	Provides architectural guidance and technical enablement to the teams on ART. They have a view on the whole system and helps in defining major components and interfaces for the system.
Product Management	They are the key stakeholders of ART. The epic owner together with the product manager consist of a team that has shared responsibilities.
Product Manager:	Owns, defines, and prioritizes the program backlog. They are scaled product owners of ART
Shared Services	Shared Services helps the team with specialty functions that cannot be dedicated to ART such as database administration, business analysis.

Team level

This is the lowest level in the SAE. It consists of the agile teams consisting of a small group of dedicated individuals, who are responsible to define the features into user stories, implement and test them in a short time box [26]. The agile teams within the ART are organized based on features or components [25]. The supporting roles ensure that teams are capable of defining, developing, testing, and delivering working solutions at least every iteration. Each team will have its own tasks tracked in their team's backlog at the team level to deliver their goals, hence delivering value at the program level [26]. Table 6 includes the roles and responsibilities within the agile teams that power the ART.

Table 6: Roles and Responsibilities at Team level [25] [26]

Scrum Master:	Runs the team meetings, manages impediments, drives the agile behavior and attends scrum of scrum meetings.
Product Owner:	Acts as the owner of the product, acts as the customer for developer options, prioritizes work and works with the Product Management to plan the PI Planning meeting.
Business Analyst:	Coordinate between the Product Owner and the development team, helps in defining user stories and acceptance criteria.
Agile Team:	They include developers, testers, and other specialists who help in refining and implementing user stories.

3.3 Requirements Engineering in SAFe

This section describes on how the RE process takes place in SAFe. Dean Leffingwell describes, how agile software requirements can be handled in large organizations using SAFe [27]. Since there are no academic articles or publications on this yet, I have made an attempt to analyze the RE process by referring to the SAFe books [26] [27], SAFe 4.5 white paper [34] and SAFe website [25].

3.3.1 RE process in SAFe

SAFe provides a guideline to enterprise organizations to deliver requirements [27]. SAFe uses a Kanban system at the portfolio level that helps in visualizing, analyzing, prioritizing and managing the flow of requirements, starting from an idea to implementation and completion [26].

The RE process in SAFe starts at the portfolio level and continues to the program and team level as shown in the figure 6. It recommends a set of sequential planning to define requirements, termed as epics, features and stories at different SAFe levels. Since SAFe does not have a step by step RE process diagram, an analysis has been made at each level by referring to the author's SAFe books [26] [27]. The requirement activities at the portfolio level is shown in red, activities at program level is shown in green and yellow at the team level.

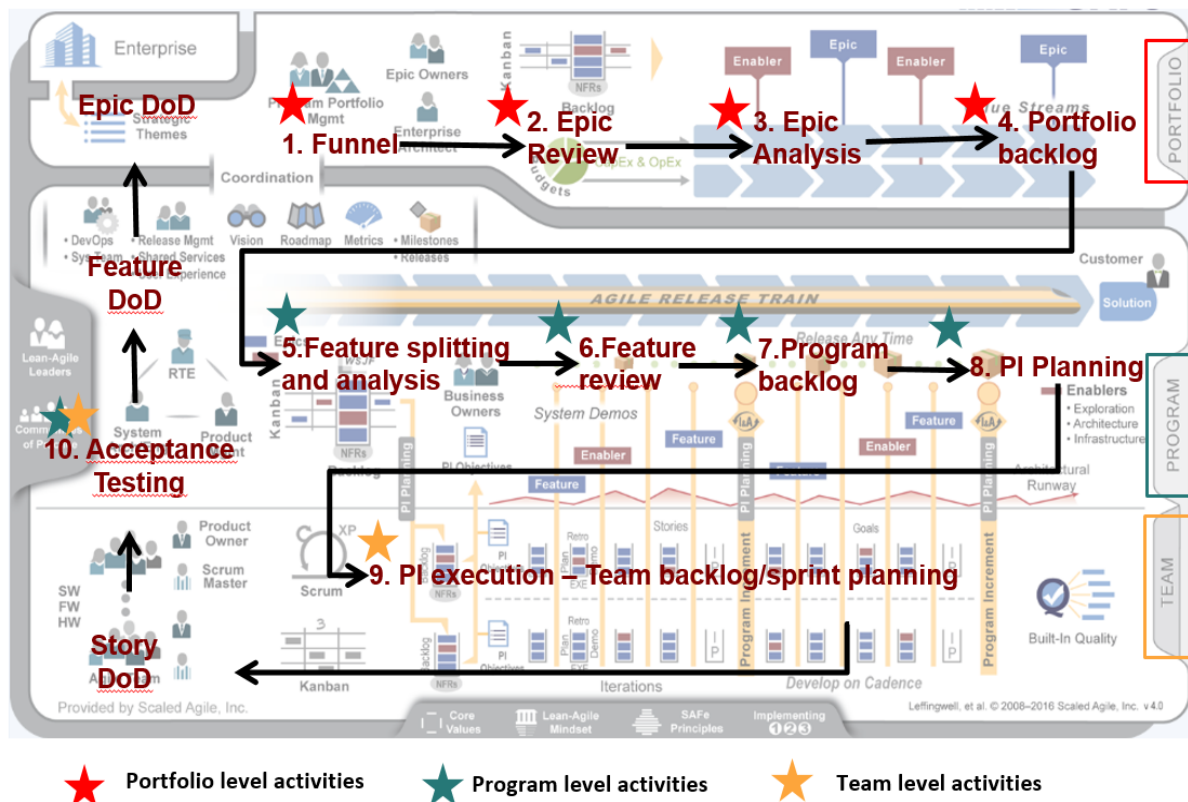


Figure 6: Analysis of RE process in SAFe levels

The requirement engineering activities at portfolio level as shown in the figure 6 includes:

1. **Funnel:** All big ideas from stakeholders in a portfolio are being captured and are welcome here [25]. They are represented as epics. However, the business ideas are derived from the strategic theme of the organization.
2. **Epic Review:** Possible epics added in the prior phase is being reviewed against opportunity, effort and cost of delay [25]. Epics are reviewed to assess its business value, return on investment, success criteria, and leading business indicators [40].
3. **Epic Analysis:** Epics are further analyzed to establish business outcome, impacts, viability, lean business case (lightweight business case) and approval process as Go/No Go decision [25].
4. **Portfolio Backlog:** Contains prioritized and approved epics by Lean Portfolio Management (LPM). Prioritized epics are added under specific ARTs in Program level. The epics are approved by communicating with the term DoR (Definition of Ready).

At the **Program level**, the epic owners and product managers splits the epics into features and transition the ownership to ARTs [27]. They simultaneously start preparing for the PI planning at program level with the respective ARTs. The program level activities include:

5. **Feature splitting and analysis:** Features are derived from the portfolio epics [27]. The features are explained in detail by the epic owner which are further divided to user stories by the product owner. The System Architect participates for providing his analysis on the architectural requirements and non-functional requirements [27].
6. **Feature Review:** The epic owner and product owner from the ART reviews the features and the acceptance criteria to ensure that the feature is ready to be included into PI Planning [27].
7. **Program Backlog:** The approved features are stored and prioritized in program backlog by the epic owner [27]. At this stage, SAFe recommends, the product owner and business analyst at team level to create user stories and refine the feature backlog regularly.

Program Increment (PI) is an integral part of agile release train (ART) which acts as an iteration at program level in Scaled Agile Framework [26]. It is driven first by having a planning meeting and then executing the planned objectives.

8. **PI Planning:** It is a face-to face event which includes stakeholders from all levels [25]. Each ART has its own PI planning, where the PI objectives are defined for the teams. The PI planning workshop will be explained in more detail in the next section. After the PI planning, the product owners and business analysts keep the user stories prepared for the teams, which are derived from the features. The user stories are approved and stored in team backlog in a DoR (Definition of Ready) state [27].
9. **PI execution:** It includes implementing and executing the PI objectives on team level by agile teams following the agile practices, such as those provided by scrum [36]. It starts by having a sprint cycle that includes a set of activities such as, sprint planning, development, testing and demo at the team level. In general, a sprint cycle spans across for 2 weeks and in total for a PI there are 4-6 sprint cycles, including IP sprint [26]. The agile team including scrum master, product owner, business analyst, developers and testers participates in the sprint planning. During the sprint planning, the team reviews the backlog, selects the user stories and estimates the tasks [26]. The user stories are implemented and then validated at the end of the sprint by having a sprint demo. This continues for each sprint and the agile teams effectively deliver the backlog items.
PI Inspect & Adapt: SAFe recommends a retrospective meeting at the end each PI known as Inspect & Adapt. It is a significant event where the current state of the solution is demonstrated and evaluated by the agile release train [26]. The result is a set of improvement backlog items for the next PI Planning meeting [25]. Hence, improving the agile release train every PI.
10. **Acceptance testing:** Acceptance tests are functional tests that verify that the system implements the story as intended [27]. To avoid large volume of manual tests they are automated wherever possible. Acceptance tests are conducted to determine if the requirements of the user stories are met [27]. Acceptance tests are done for user stories at team level and for features at program level.

3.3.2 Epics, Features and Stories

SAFe suggests to demonstrate the system behavior through epics, features and user stories [27] [44]. The figure 7 depicts the sequential split of epic, feature and user stories at each SAFe level. In this section, we describe in detail to understand an epic, feature and user story and how are they linked.

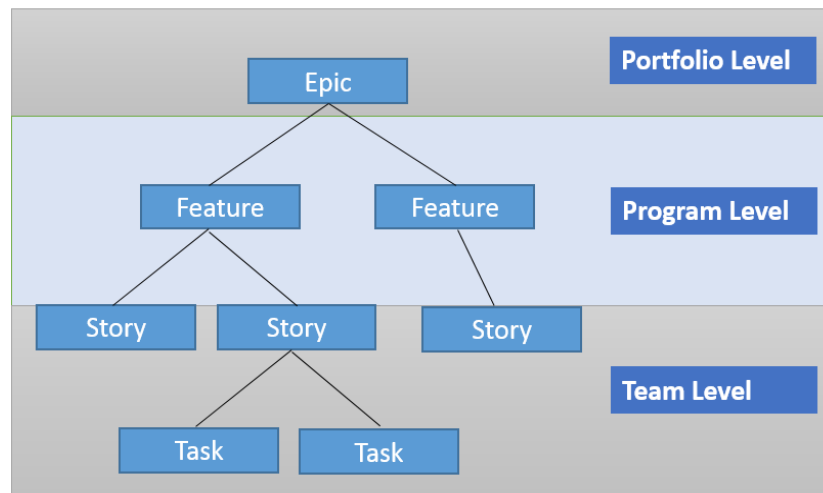


Figure 7: Sequential split of epic, feature and user stories at SAFe levels

Epics are the starting point of requirements in a large enterprise organizations and are defined at portfolio level [27]. It can be a business idea or an enterprise initiative derived from strategic theme of the organization. The epics are analyzed and managed in the Portfolio Kanban system and is managed by Lean Portfolio Management (LPM) [27]. In addition to the LPM it also requires participation from other stakeholders such as epic owners and enterprise architects to address the flow of epics. Epic can be of two types, business epics and enabler epics. Business epics provides a business value and enabler epics contributes to an architectural runway [25]. Portfolio Kanban is used for managing the epics such as funneling, reviewing, analyzing and approving an epic [26]. After a thorough review process, an approved epic contributes to a lean business use case [27]. Epic is further divided into multiple features and can span across more than one team.

The authors in the article [42] defines epic as a set of user stories, whereas the authors in the article [43] defines epic as a theme or goal that is often broken down into multiple features and can span across more than one team. The concept of epic defined as a goal and breaking down into features is similar to what SAFe defines. In SAFe, epics are prioritized at portfolio level in epic backlogs, which are later broken down into features and are implemented through multiple Program Increments [27].

Features are derived from epics and each feature is implemented within a program increment. A feature can be defined as a service that has been requested by a stakeholder [27]. Each feature can have a business hypothesis statement and a detailed acceptance criteria. Product managers

usually takes the responsibility of managing features along with product owners. Each program increment prioritizes the features based on either weighted shortest job first (WSJF) or by assigning manual business values [27]. Features are estimated and, epic sizing can be derived by summing the individual feature estimates [25]. Features are further divided into stories.

Stories represent a small description of a desired functionality written from user's perspective [25]. They are derived from a feature at program level as shown in the figure 7. A story, also called as user story, is a small piece of business value that can be achieved in an iteration or sprint [41]. Ideally, a user story should have enough information that enables to test and satisfy the customer. Each user story can be further divided into tasks across the agile team members such as developers and testers. Table 7 summarizes and provides a clear difference between epic, feature and user story. It helps in attaining a good understanding based on description, SAFe levels, prioritization, delivery, timeframe, testable and acceptance. This analysis was done by referring to the SAFe books [26] [27] and website [25].

Table 7: Discriminating Epics, Feature and Stories

	Epic	Feature	Stories
Description	Large step changes in corporate capability; a business idea or business requirement	Business hypothesis statement derived from epic and acts as a service that fulfills user needs	Something of value a team can complete in a sprint. It is derived from a feature.
SAFe level	Portfolio	Program	Team
Prioritization	Prioritized in portfolio backlog by Lean Portfolio Management(LPM) and epic owner	Prioritized in program backlog by epic owner or product manager	Prioritized in team backlog by Product Owner
Delivery	Delivered by implementing features across program increment (PI)	Delivered by implementing features by a single program within single PI	Delivered by implementing tasks by a single agile team within a single sprint
Time frame & sizing	6-12 months	Fits in one PI (8-12 weeks)	Fits in one sprint (2 weeks)
Testable	No	Yes	Yes
Acceptance	By LPM and epic owner	By product manager or epic owner	By product owner

Some important points observed from table 7 is that, the features and user stories are testable and not epics. This means, once the user stories and features are tested and accepted by the product owner and epic owner, they fulfill the epic which is the business requirement. At this stage, the LPM and epic owner approves the epic with the term Epic DoR (Definition of Done). Another point that can be observed is that, the epic owners participate both the portfolio and program level to prioritize and accept the epics and features.

3.3.3 Good RE Practices of SAFe

Effectively implementing a new set of RE practices in a project team, program or enterprise is not easy. In addition, further 'leaning' the organization often requires eliminating or reducing requirements specifications, design specifications, sign-offs and etc. [27]. However, the suggestion to adopting to good RE practices of SAFe is not new in large organizations working with large number of teams [25].

SAFe recommends several practices that can be considered for RE process in large organizations using SAFe. In this section, an analysis is done on identifying good RE practices by focusing on the RE process of SAFe from the books of Dean Leffingwell. These books are: Scaling Software Agility: Best Practices for Large Enterprises (2007) [45] and Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (2010) [27]. In addition to this the official SAFe website was also referred [25]. Table 8 summarizes the good RE practices of SAFe which can natively scale to enterprise level to improve the RE process.

Table 8: Good RE practices of SAFe

Name of the Practice	Short Description	Reference
Vision	The vision describes the stakeholder's view of the solution to be developed in terms of stakeholders needs.	[27], [45]
Roadmap	Establishes alignment across all the teams in ART while also providing predictability to the deliverables over an established timeline horizon	[27], [45]
Collaborate Planning	Encourages all stakeholders to come together during the planning phase. The collaborative efforts increase visibility, loyalty, and acceptance and buy in from all stakeholders.	[27]
Requirements Discovery toolkit	Practice which is used to better understand what needs to be built and why, by applying techniques such as, brainstorming, interviewing, using mock ups and etc.	[27]
Well defined epics, features and stories	Use templates and specifications to define epics, features and stories with a detailed acceptance test.	[27]

Domain Modelling	It is a visual representation of the real world entities and their relationships that cover the problem domain.	[25], [23], [27]
Model Based Systems Engineering	Is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities	[25]
Manage epics using Kanban systems	Kanban Systems are used for visualizing workflow, limiting the work in progress, measuring and managing the flow of epics.	[27], [45]
Maintain product backlog	The product backlog is a repository for all the upcoming work which is anticipated to be delivered. SAFe encourages to use backlogs on all SAFe levels - portfolio, program and team.	[27]
Organize agile teams at scale	Agile teams organized in ART at program level should align to a common mission to achieve common business goals.	[27]
Program Increment (PI)	It is a scaled iteration at program level which includes a scaled sprint planning known as PI Planning. PI planning, is a face-to-face conversation to convey information.	[25], [27]
Manage feature dependencies	It is a practice applied during PI planning meeting to track and manage feature dependencies across agile teams in ART in a form of a program board.	[25], [27]
PI Inspect & Adapt	It is a retrospective meeting at the end each PI where the current state of the solution is discussed and evaluated in ART. The result is a set of improvement backlog items for the next PI Planning meeting.	[27]
Scrum of Scrums	An agile practice which scales scrum to program level. Scrum masters of each team identify interdependencies, report status and state any risks and impediments.	[45]
Well defined roles and responsibilities	A complex framework like SAFe, requires well-defined roles and responsibilities to have a well-organized RE process in large organizations.	[27]
Community of Practice (CoP)	It is a practice of sharing knowledge, by creating groups of people who share interest in a common topic.	[25], [27], [39], [48]
Managing distributed teams	Large corporates are distributed. They should be managed with proper communication and the necessary networking and tooling architecture	[45]

Some good RE practices of SAFe are selected from table 8 and are described further in detail. These selected practices needs more detailed explanation and can be used as potential good RE practices from the perspective of empirical study.

Requirements Discovery toolkit: This is a practice which is used to better understand what needs to be built and why. There are variety of software requirements techniques which are recommended and can be used by teams, for example, brainstorming, interviews and questions, spikes, use-case modeling, mock-ups, etc. [27]. These techniques in turn can be applied as good RE practices for requirements elicitation and analysis. In this context, there have been identified, two best practices that can be used to improve the RE process.

Domain Modelling: It is a visual representation of the real world entities and their relationships and responsibilities that cover the problem domain [23] [25]. Since, there is usually a gap in understanding the problem domain or interpreting the requirements, domain modelling helps in envisioning the solution and in resolving ambiguities in requirements [27]. It can be represented as a UML diagram or Entity-Relationship diagram [37]. In SAFe, domain modelling can be done for the backlog items at portfolio, program and team level.

Domain modelling can be continuously used to support [25]:

- Analysis of epics at program level
- Design workshops at different levels
- In refining vision or roadmap, during preparation of program increment

Domain modelling is a continuously refactored as and when the enterprise knowledge about the domain evolves. Requirements and domain modelling are interlinked to each other. Domain modelling helps in shared understanding of the requirements and requirements help in creating and clarifying the model [25].

Model Based Systems Engineering (MBSE): It is a practice recommended by SAFe that relates real world entities and relationships to add more clarity to requirements while dealing with complex systems.

According to SAFe [25] “*MBSE is the application for modelling requirements, design, analysis, and verification activities as a cost-effective way to explore and document system characteristics*”. These models help in learning their properties and behavior by validating in an early stage, hence enabling fast feedback on requirements and design decisions. This practice supports SAFe principle 4 ‘*Build incrementally with fast, integrated learning cycle*’ [25].

From the above definition, it can be analyzed that the practice of applying modelling helps in exploring the structure and behavior of the system, thus improving communication by enhanced knowledge and receive faster feedbacks, than any other costlier methods.

Software Kanban systems: SAFe suggests the development and implementation of Kanban systems for business and architectural portfolio epics. The Kanban system describes four queues that an epic passes through on the way to implementation: Funnel, Backlog, Analysis, and Implementation [45]. The Kanban Systems are used for visualizing workflow, limiting the work in progress, measuring and managing flow, making process policies explicit and using models to recognize improvements [27].

Program Increment (PI): is an integral part of agile release train (ART) which acts as an iteration at program level in Scaled Agile Framework [26]. It is driven first by having a planning meeting and then executing the planned objectives. **PI Planning:** It is a face-to face event which includes stakeholders from all levels [25]. Each ART has its own PI planning, where the PI objectives are defined for the teams. The PI planning workshop will be explained in more detail in the next section.

Program Board to manage feature dependencies: It is a physical display that highlights the new feature delivery dates, feature dependencies among teams in ART or other ARTs [27]. The program board is one of the primary outputs of a successful PI planning meeting [25]. This will be explained in more detail in the next section of Program Increment.

Communities of Practice (CoP): are organized groups of people who collaborate regularly to share information [25] [48]. These groups share a common interest in the specific topic such as technical or business domain. SAFe uses this concept as a practice to enable practitioners to exchange knowledge and skills with people across entire organization [25]. It helps in overcoming the common problems in teams such as knowledge gap in domain and system applications. Community of practice is viewed as a social learning system [39]. Wenger [49] categorizes CoP into three different traits which is similar as to how SAFe also categorizes [25]:

- Domain – Business domain or any shared interest.
- Practice – A group of people following a process or body of knowledge, experiences or techniques.
- Community –A selected group of people who want to participate on a specific topic and gather individually.

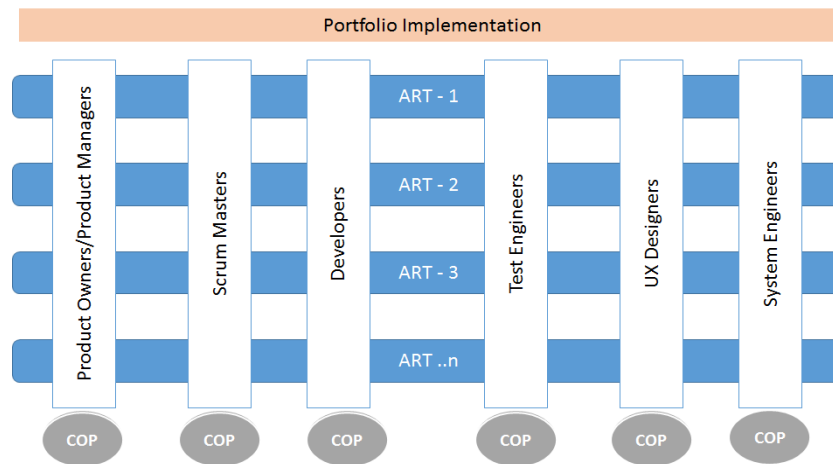


Figure 8: Role based community of practice (CoP) [25]

SAFe recommends to have a CoP for the different roles across ARTs as shown in figure 8. The roles include, Product Owners/Product Managers, Scrum Masters, Test Engineers, Developers, UX Designers and System Engineers. These *role* based CoPs, gather regularly to share their experiences, knowledge, concerns and gaps to avoid any risks to project delivery and enhance knowledge within the roles [26]. Paasivaara and Lassenius suggest eight characteristics for a successful CoP [48]. The eight characteristics are: interesting topic with concrete benefits to participants, passionate leader, proper agenda, decision making authority, open community, supporting tools to create transparency, suitable rhythm, and cross-site participation when needed.

Good RE practices for agile development teams in SAFe

After identifying some good RE practices of SAFe, further analysis was done on good RE practices for agile development teams in SAFe. The intention of this analysis was to verify if these practices can also be applied in SAFe. The table 9 shows the list of good RE practices for agile development teams which are verified against the column of SAFe with necessary references.

Table 9: Analyzing good RE practices for agile development teams in SAFe

Good RE practices for agile development teams	Applied in SAFe	Reference
<i>Face to Face Communication</i>	Yes	[25] [27]
<i>Prioritize requirements</i>	Yes	[25] [27]
<i>Modelling and Prototyping</i>	Yes	[25] [27] [45]
<i>Use peer reviews, scenarios and walkthrough to validate and verify requirements</i>	Yes	[25] [27]
<i>Test driven development</i>	Yes	[25] [27]
<i>Iterative RE</i>	Yes	[25] [27] [45]
<i>Train team members to use right processes, tools and technology</i>	Yes	[25], [27]

The results in table 9 show that, almost all the good RE practices for agile development teams can be applied in SAFe. SAFe framework uses these good RE practices which are used for agile development teams and scales the practices from team level to program and portfolio level. For example: the face-to-face communication is a practice also applied in SAFe at program level during PI planning meeting.

Each level in SAFe contains practices which when adopted collectively leads to significant improvements in scaled agility [25]. Hence, it is necessary for the large organizations to adopt to good RE practices on the lower levels first and then continue at the next levels. This is because, the RE practices on the higher level are dependent on the practices used at the lower level.

3.3.4 Program Increment (PI)

Program Increment is an integral part of Scaled Agile Framework. It is defined as “The most efficient and effective method of conveying information to and within a team with face-to-face conversation” [25]. According to Larman Craig and Bas Vodde [28] when there are people with larger teams in a company, it is useful to convey a consistent introductory message to everyone. With the PI planning meeting, SAFe takes the face-to-face conversation to a higher level and convey the message to everyone. This takes place at the program level, where all the stakeholders and agile teams in ART come together to plan and commit to deliver the features.

Program Increment is preceded by a planning meeting which is a synchronized routine of two day event that occurs every 8-12 weeks. Figure 9 depicts the typical agenda of a PI planning meeting.



Figure 9: PI Planning agenda [25]

PI planning is a cadence-based, face-to-face event that serves as a pacemaker to an agile enterprise, aligning all the teams on the ART to a common goal and vision [25]. All stakeholders, members who are involved in the train attend this event personally if possible or connect remotely. However, in geographically distributed ARTs, the event may occur at multiple locations simultaneously, with real-time communication between the locations. The result of this event is to achieve a common understanding of customer requirements and goals, and to make a commitment to an agreed set of objectives for the next PI [26]. It is reported in many case studies that, the PI planning meeting has been a major success factor for adopting SAFe [31].

PI planning requires a good preparation, coordination and communication [26]. Prior to the PI planning meeting, the product management collaborates with the customer, other stakeholders and product owner to develop the roadmap, program vision, prepare and estimate the features and milestones. System Architect/Engineering prepares technical briefings and guidance to support planning. Altogether they validate the feature list and set expectations for the PI planning meeting [26]. The table 10 is derived from the agenda of the PI planning (figure 9). It gives an overview of what results are achieved during the PI Planning meeting.

Table 10: Results of PI Planning meeting

Input	Product or Solution Vision, Roadmap and Top 10 features
Day 1 results	<ul style="list-style-type: none"> • The business context and upcoming objectives are discussed • The product/solution vision is discussed and prioritized features • Teams develop draft plans during team breakouts and identify the risks and dependencies. Features are broken down into stories (sometimes this is done before the PI planning meeting by the product owner and business analyst). • Architects and Product managers circulate around different teams during team breakouts. • The teams present draft plans, risks and dependencies. • The program board is used to show dependencies across teams in an ART. • The management reviews and makes adjustments based on challenges and risks
Day 2 results	<ul style="list-style-type: none"> • Planning adjustments are made based on the previous day's management meeting • Again during team breakouts the teams develop final plans and refine risks and challenges. During this time, the business owners circulate and assign business value to team objectives • Teams present final plans, risks, and dependencies. • Remaining program-level risks are discussed in the auditorium where all team members are present • Team and program confidence vote is taken
Output	Committed PI Objectives and Program Board

The output of a successful PI Planning meeting are committed PI objectives and a Program Board. **PI objectives** is a summary of business and technical objectives that are created by each agile team in ART, which they intend to achieve in the upcoming PI [25]. The objectives have the business value assigned by the epic or business owners.

According to Eric Willeke, who is a SAFe Program consultant trainer, states that the main qualities of PI objectives are their ability to [25]:

- Validate understanding of Intent
- Focus alignment on outcomes rather than process
- Summarize data into meaningful and steerable information

Program Board is a simple physical display that provides a big picture of any feature dependencies across the teams in an agile release train [25]. This occurs during the PI Planning meeting. After the team breakouts, each team drafts plans to their given features and recognize the risks and dependencies [25]. These risks and dependencies are brought in together and the Scrum master or the Product owner takes lead to discuss this with the other teams and the management. As part of this discussion, the dependencies are put on the program board across the team names. The figure 10 represents a program board which helps the stakeholders to understand the underlying gaps and dependencies to plan the work accordingly.

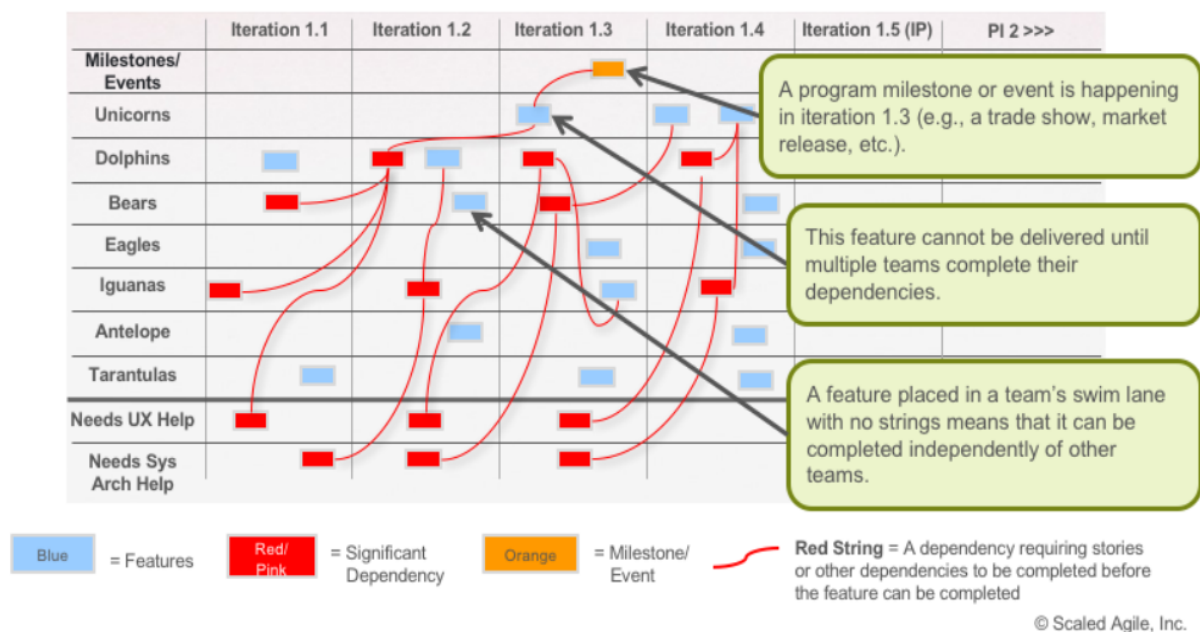


Figure 10: Program Board [25]

This is a unique method which SAFe believes that it enables communication face-to-face across teams and help in managing risks and dependencies at the planning phase itself [27]. However, this makes it challenging for the globally distributed teams as they are not co-located.

3.4 Summary

This section summarizes and combines the main results of the literature study. The literature study was divided into three sections: Requirements Engineering (RE); Scaled Agile framework (SAFe); and Requirements Engineering in SAFe. It attempts to provide a deeper knowledge on requirements engineering activities and good RE practices. The goal of the literature study was to analyze the RE process in SAFe and identify some good RE practices that can be applied to improve the RE process.

Analyzing RE process in SAFe

In order to substantiate my research problem on how to improve the RE process using SAFe in globally distributed teams. It was first important to understand the term RE, analyze the typical RE activities and its good practices. Next, it was important to understand what is SAFe and big picture of the SAFe model. The results of Requirements engineering section and Scaled Agile Framework section, acted as an input to further analyze the RE process in SAFe and identify good RE practices of SAFe.

Requirements engineering (RE) is a term, which is used to describe the process of creating requirements for a system [13]. It was analyzed that requirements engineering is one of the initial steps in software project model that needs to be well defined, understood and maintained in any software development organization. Requirements engineering consists of systematic and repeatable activities known as RE activities that ensure the completeness, consistency and relevance of the system requirements [3]. The RE activities are requirement elicitation, requirement analysis, requirement representation, requirement validation and requirement management. Requirement elicitation is the first step in RE process to discover customer and user needs. It uses techniques that can also be used in SAFe such as, questionnaires/surveys, group discussion, scenario-based discussions, whiteboard sessions, interviews, prototyping and goal-based discussions [17]. Some of these techniques can be effectively used in SAFe while defining an epic, feature and user story. Also, the RE activities acted as a good input to analyze the RE process in SAFe.

SAFe is a framework that uses agile methods and practices for implementing software for large enterprises [26]. It consists of three levels: portfolio level, program level and team level. Since, SAFe is a complex framework to understand, a Big Picture of SAFe model [25] was used in the literature study. It provided a visual representation of the framework by highlighting the levels and important roles for the RE process. Understanding the big picture of SAFe with important roles and responsibilities helped in analyzing the RE process in SAFe.

The RE process in SAFe starts at the portfolio level and continues to the program and team level. An analysis was done on identifying the RE activities at each SAFe level. It recommends a set of sequential planning to define requirements, termed as epics, features and stories at different SAFe levels [26]. The RE activities at the portfolio level are:

- **Funnel:** All big ideas from stakeholders in a portfolio are being captured and are represented as epics.
- **Epic Review:** Epics are reviewed to assess its business value, return of investment, success criteria, and leading business indicators [40].
- **Epic Analysis:** Epics are further analyzed to establish business outcome, impacts, viability, lean business case (lightweight business case) and approval process as Go/No Go decision [25].
- **Portfolio Backlog:** It contains the prioritized and approved epics by Lean Portfolio Management (LPM) team which will be added under specific ARTs in Program level.

The RE activities at portfolio level are:

- **Feature splitting and analysis:** Features are derived from the portfolio epics [27]. The features are explained in detail by the epic owner which are further divided to user stories by the product owner.
- **Feature Review:** The epic owner and product owner from the ART reviews the features and the acceptance criteria to ensure that the feature is ready to be included into PI Planning [27].
- **Program Backlog:** The approved features are stored and prioritized in program backlog by the epic owner [27].
- **PI Planning:** Each ART has its own PI planning. It is a face-to face event which includes stakeholders from all levels, where the PI objectives are defined for the teams.

The RE activities at team level are:

- **PI execution:** It includes implementing and executing the PI objectives by agile teams. It starts by having a sprint cycle such as, sprint planning, development, testing and demo at the team level. During the sprint planning, the team reviews the backlog, selects the user stories and estimates the tasks [26]. The user stories are implemented and then validated at the end of the sprint by having a sprint demo.
- **Acceptance testing:** Acceptance tests are functional tests that verify that the system implements the story as intended [27]. Acceptance tests are done for user stories at team level and for features at program level.

SAFe uses a Kanban system at the portfolio level that helps in visualizing, analyzing, prioritizing and managing the flow of requirements, starting from an idea to implementation and completion [26]. The RE activities at portfolio level helps in defining relevant high level business requirements represented as epics that needs to be implemented. This can be correlated to typical RE activities used in agile development projects, that creates definition of business outcome to be achieved in a development process.

The authors in the article [42] defines epic as a set of user stories, whereas the authors in the article [43] defines epic as a theme or goal that is often broken down into multiple features and can span across more than one team. The concept of epic defined as a goal and breaking down into features is similar to what SAFe defines.

At the program level, the epic owners and product managers splits the epics into features and transition the ownership to ARTs [27]. Simultaneously, the epic owners and product managers start preparing for the PI planning at program level. Program Increment (PI) is an integral part of agile release train (ART) which acts as an iteration at program level in Scaled Agile Framework [26]. It is driven first by having a planning meeting and then executing the planned objectives.

In SAFe, epics are prioritized at portfolio level in epic backlogs, which are later broken down into features and are implemented through multiple Program Increments [27]. Also a table was created to summarize and provides a clear difference between epic, feature and story. It helped in attaining a good understanding based on description, SAFe levels, prioritization, delivery, timeframe, testable and acceptance. It was observed that the features and user stories are testable and not epics. This means that, once the user stories and features are tested and accepted by the product owner and epic owner, they fulfill the epic which is the business requirement. At this stage, the lean portfolio management and epic owner approves the epic with the term 'Epic DoR' (Definition of Done). Another point observed was that, the epic owners participate both at the portfolio and program level to prioritize and accept the epics and features.

Good RE practices that can be used in SAFe

As part of this research study, it was important to identify what are the good RE practices that can be used in SAFe to improve the RE process in globally distributed teams. There are not many articles related to this, or of their related challenges and benefits. Hence, the practices were analyzed and selected based on the RE process of SAFe and by referring to the SAFe books [27] [45].

SAFe recommends several practices that can be considered for RE process in large organizations. In addition to these good RE practices of SAFe, further analysis was done on good RE practices for agile development teams in SAFe. The intention of this analysis was to verify if the team level practices can be scaled and be applied in large organizations using SAFe. The results show that, almost all the good RE practices for agile development teams can be applied in SAFe. SAFe framework uses these good RE practices which are used for agile development teams and scales the practices from team level to program and portfolio level. For example, the face-to-face communication is a practice also applied in SAFe at program level during PI planning meeting.

The challenge is in large organization where it is necessary to coordinate and communicate between several agile teams, and also between different organizational units [35]. Adopting RE practices in large organizations have more dependencies and risks, it requires change to the entire organizational structure [32]. However, each level in SAFe contains practices which when adopted collectively leads to significant improvements in scaled agility [25]. Hence, it is necessary for the large organizations to adopt to good RE practices on the lower levels first and then continue at the next levels. This is because, the RE practices on the higher level are dependent on the practices used at the lower level.

The practices are adopted at different levels in SAFe. To describe in a nutshell, SAFe adopts the scrum practices at the team level. At the program level, it consists of agile release train and program increment, which is analogy to sprints at the team level working in a larger time frame. The scope is broaden at the portfolio level, where the requirements gathering is done and are represented as epics, defining the large development initiatives. The good RE practices that can have a high potential to be applied in SAFe, in the empirical study of this research are:

- *Requirements discovery toolkit*: is a RE practice which is used to better understand what needs to be built and why, by applying techniques such as, brainstorming, interviewing and using mock ups [27].
- *Domain modelling*: Visually representing requirements defining the real world entities and their relationships that cover the problem domain [23] [27]. Domain modelling helps in envisioning the solution and interpreting the requirements. It can be represented as a UML diagram or Entity-Relationship diagram.
- *Maintain product backlog*: The product backlog is a repository for all the upcoming work which is anticipated to be delivered. SAFe encourages to use backlogs on all SAFe levels - portfolio, program and team [27].
- *Organize agile teams at scale*: Agile teams organized in ART at program level should align to a common mission to achieve common business goals [27].
- *Program Increment*: It is a scaled iteration at program level which includes a scaled sprint planning known as PI Planning. PI planning, is a face-to-face conversation to convey information across teams in an agile release train (ART).
- *Manage feature dependencies*: It is a practice applied during PI planning meeting to track and manage feature dependencies across agile teams in ART in a form of a program board [25] [27].

- *Community of Practice (CoP)*: It is a practice of sharing knowledge, by creating groups of people who share interest in a common topic [25]. The groups can be created based on roles such as, Product Owners/Product Managers, Scrum Masters, Test Engineers, Developers, UX Designers and System Engineers. These role based CoP, gather regularly to share their experiences, knowledge, concerns and gaps to avoid any risks to project delivery and enhance knowledge within the roles [26].
- *Managing distributed teams*: Large corporates are distributed. They should be managed with proper communication and the necessary networking and tooling architecture [45].

4. Empirical Study

4.1 Analysis of the case company project

Background

Case company project - ‘Project X’ train has grown rapidly during the last year and at the moment the train has good scope for main development areas in different applications. To do so, it aims to follow the SAFe concepts.

The train also termed as agile release train (ART) in SAFe has 8 agile teams consisting of approximately 100 members both onshore and offshore. The train has a routine of 8-12 week program increment which includes - planning, development, testing and retrospective cadence, and implements continuous product development flow. Similarly, each train in the case company has similar structure and dedicated resources necessary to continuously define, build, and test valuable system level solutions.

The agile team is a cross-functional group of members distributed between Finland, Poland and India. The members have the ability and authority to define, build and test, all in a short iteration time box. As such, the team includes developers and testers, a scrum master, business analyst and product owner, which are necessary to deliver successful results. Teams operate in the context of the business requirements, and architectural guidance of the train, each doing their part, collaborating with other teams, and participating in key release train events.

4.1.1 Study of RE process using SAFe

In this section we will mainly be focusing and analyzing on how RE process is carried out in the case company project using Scaled Agile Framework. This is done by participating in the case company project, making observations and conducting interviews.

Program Increment (PI)

Program Increment is an 8-12 week iterative process of agile software development life cycle, starting from the RE process, development, testing and deployment. This takes place mainly at the Program and Team level coordinating together with the Portfolio level. Since my research study is based only on the RE process, my focus would be to concentrate only on the events that connect with RE process. In the empirical study, my analysis of the case company project is divided into three phases. As shown in figure 11, the three phases are: Pre-PI Planning, PI Planning and PI Execution

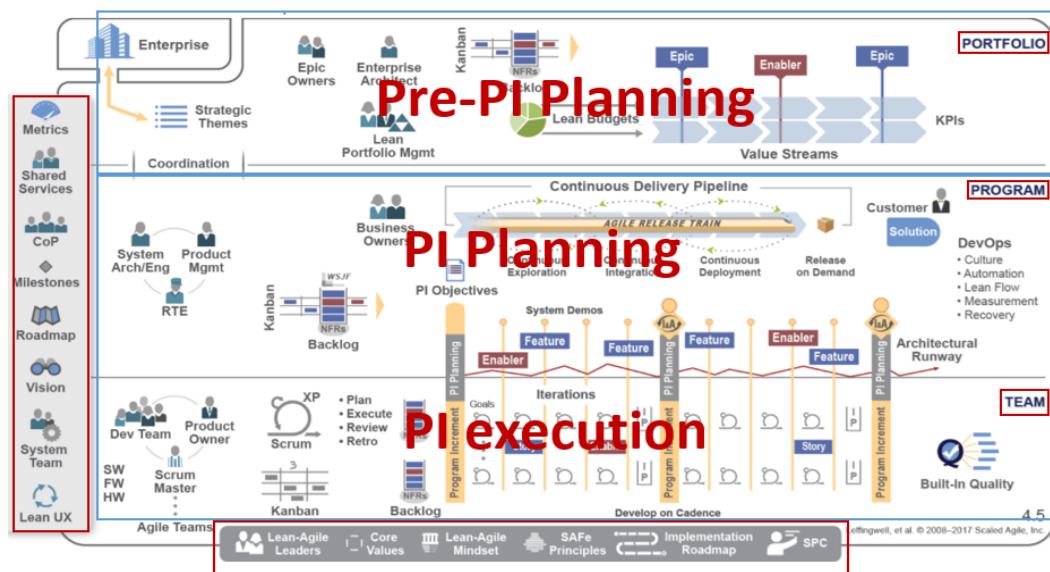


Figure 11: Phases of program increment at SAFe levels (SAFe Big picture [25])

1. Pre-PI Planning

Prior to the PI Planning meeting, the product management collaborates with the customer, other stakeholders and product owner to develop the roadmap, program vision, and prepare and estimate the epics, features and milestones. This was done by having face to face conversations and applying requirement elicitation techniques, such as, group discussion, scenario-based discussions, whiteboard sessions, interviews and prototyping. In the SAFe framework, the initial activities of requirements definition such as requirement elicitation and analysis takes place at the portfolio level which includes the business owners, epic owners and enterprise architect.

The portfolio level activities focus on discovering new ideas. New ideas can also be called as business requirements which bring business value. The program portfolio management and epic owners work together on eliciting the new ideas. The epic owner writes the epics and later splits them into features. In the current case company project, all the activities here take place at the onshore location which is at the customer location. Once the features are ready, they are prioritized and allocated to different teams in the train at the program level. Table 11 provides a detailed understanding of some important roles and responsibilities that take place in RE process at portfolio level.

Table 11: Roles and Responsibilities at portfolio level before PI Planning

Level	Role	Location	Responsibilities
Portfolio	Lean Portfolio Management	Client side (onshore)	<ul style="list-style-type: none"> ➤ Establish strategic themes, manage Epics, make forecasts, empower local decision-making
Portfolio	Epic Owner	Client side (onshore)	<ul style="list-style-type: none"> ➤ Builds Business cases and write epics for the development ideas. ➤ Progress Epics in Portfolio Kanban system ➤ Participates in the entire life-cycle of Epic – from an idea to measuring the customer benefits ➤ Presents the development ideas for prioritization and decision making ➤ Kicks off the Epic implementation and follow-up
Portfolio	Enterprise Architect	Client side (onshore)	<ul style="list-style-type: none"> ➤ Recognizes the constraints and possibilities ➤ Supports Business Epic definition; steers the portfolio level solutions by supporting business in preparation and ensures that the planned solutions are in line with the business targets and technology choices

Program level activities drive to plan what features are to be developed and released into production next. Additionally the solution is validated with the customer to ensure that objectives are met. Since it is an iterative process, after the business goals and objectives have been delivered, the train will continue to work with new set of business goals that comes from the customer. In the current case company project, all the activities here take place at onshore location some at the client side and some at the company side.

Program level activities include:

- Features are derived from Portfolio level epics
- The features are explained in detail by the epic owner which are further divided to user stories by the Product Owner.
- The System Architect participates for providing his analysis on the architectural requirements and contribute to the planning. He is also responsible to analyze whether NFR's require new user stories or test cases or whether they influence the acceptance criteria of some stories as constraints
- Epic Owner reviews the features and the related documentation to ensure that that feature is ready to be included into PI Planning.
- Features in Program Backlog are prioritized
- Proceeds with organizing the PI Planning workshop that define PI Objectives. The PI planning workshop will be explained in more detail in the below sections.

The features are split into user stories and the product owner with the help of business analyst start to draft the user stories. This helped the teams to be prepared before the PI planning. The other benefit was that, it helped the offshore members such as business analyst, developers and testers to analyze and understand the requirements. It also helped the product owner and business analyst to identify any impediments or risks. These issues are then brought during the PI Planning meeting to the management and other teams. But this was not always the case, sometimes the features were not ready or available earlier enough to start drafting the user stories. This led the agile teams to be less prepared for the PI planning and keeping them under stress. Table 12 provides a detailed understanding of some important roles and responsibilities that take place in RE process at program and team level before the PI planning.

Table 12: Roles and responsibilities at program and team level before PI Planning

Level	Role	Location	Responsibilities
Portfolio / Program	Epic Owner/ Product management	Client Side (onshore)	<ul style="list-style-type: none"> ➤ On program level: Splits Business Epics into Features together with the Product Manager in Finnish language ➤ Ensures that the features are prioritized and is ready for PI Planning ➤ Has dialog together with teams and product management ➤ Participates in identifying dependencies and risks
Program	System Architect	Company side (onshore)	<ul style="list-style-type: none"> ➤ Works with customers, Epic owners, Product Managers and Product Owners to understand and maintain a high level understanding of the current and upcoming requirements for the system ➤ Presents the technological vision for the problem and participates actively during the planning process

Program/ Team	Release Train Engineer (RTE)	Company side (onshore)	<ul style="list-style-type: none"> ➤ Prepares and facilitates PI planning meeting ➤ Helps manage Risk and dependencies ➤ Tracks PI Feature initiation and completion ➤ Establishes the annual calendars for sprints and PI's ➤ Works with Product Management, Product Owners, and Business Owners to help assure strategy and execution alignment. ➤ Coordinates during requirement change management
------------------	---------------------------------------	------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

From the table 11 and table 12 it can be noticed that the epic owners from the client side participate actively at the portfolio level and at the program level. They in turn coordinate with the product owners from the company side who participates both at the program and team level activities. The Enterprise architect from the client side at the portfolio level collaboratively works with the system architect at the program level who represents from the company side. These roles help in coordinating the process at the different levels of SAFe framework

The team level activities include participation in PI planning and to help in executing the PI objectives (features and user stories). The PI planning and execution will be described further in the next sections. In the current case company project, all the team level activities take place both at the onshore location (Finland) and offshore location (India & Poland). Each team analyzes the features and stories allocated to them and helps in identifying risks and impediments. System Architect/Engineering prepares technical briefings and guidance to support planning. Altogether they validate the epic and feature list and set expectations for the PI planning meeting.

Therefore, most of the challenges were faced at team level and also at the program level. Since the teams are distributed across in different locations, it was very difficult to coordinate and share a common understanding of requirements.

2. PI planning

Planning is considered as an effective way of conveying information to the team in a face-to-face conversation. This makes it challenging for the teams who are located in different locations. This was a good area to do my analysis and research as I could identify some gaps in communication while the requirements were discussed and committed to deliver.

The Release Train Engineer (RTE) facilitates the PI planning meeting. The PI planning meeting in this case company project is held for 2 days in a big auditorium within the company premises where all participants can be accommodated. Also, some meeting rooms are booked outside the auditorium for team breakouts.

The PI planning meeting is arranged for the train which is held once in 3 months (8-12 weeks). The participants include members from both onshore and offshore teams and clients (portfolio management and epic owners). Since we have geographically distributed teams, the event occurs simultaneously with real-time communication between the locations. The PI Planning meeting driven by the RTE takes place mainly at the onshore location where the clients are also present. Audio setup and internal meeting links such as Skype and WebEx is setup to make sure the real-time communication is effective with the offshore teams. It is expected that the offshore teams participate actively during the event.

The two day agenda described in literature study, is followed very well according to the schedule. The day 1 starts with the business owners presenting their business context, product vision or roadmap. During the planning context session, a list of epics and features are discussed at a higher level giving their business value to it. This approach of sharing the business context and vision gives a good understanding and perspective to all the members in the project including the technical team.

The team breakouts are the longest part of the day event. They breakout out into their respective teams which in this case is 8 teams and they gather in the respective meeting rooms which are already booked for each team by the RTE. Each meeting room has an audio setup to connect to the team members working in different locations. The product owners and scrum masters lead this and during this session, the members discuss their respective features and user stories. They together validate the requirement and identify some risks and dependencies that could create impediments to their work or deliverables. The epic owners, architects and sometimes acceptance test specialists circulate around different meeting rooms to participate in the team discussion. Table 13 provides a detailed understanding of some important roles and responsibilities that take place in RE process at program and team level during PI planning.

Table 13: Roles and responsibilities at program and team level during PI planning

Level	Role	Location	Responsibilities
Program / Team	Product Owner	Company side (onshore)	<ul style="list-style-type: none"> ➤ Participates in requirement analysis and prioritization along with the epic owner, but not always. ➤ Splits features to user stories ➤ Co-ordinates with the client and the team both in onshore and offshore locations
Team	Business Analyst	Company side (offshore)	<ul style="list-style-type: none"> ➤ Shares the responsibilities of the product owner. ➤ Co-ordinates with the product owner and the team both in onshore and offshore locations
Team	Team members	Company side (onshore & offshore)	<ul style="list-style-type: none"> ➤ Participates in the PI planning to understand and validate the requirements ➤ Analyze the features, stories and help in identifying risks and dependencies at train level. ➤ commit in the PI planning

During the team breakouts, the requirements were validated by each team based on the features allocated to them. The teams identified risks and dependencies and brought them together in the auditorium and discussed with other teams. The feature dependencies were represented into the program board as shown in figure 12. It was observed that, while the dependencies were discussed with each team's product owners or scrum masters, the offshore team's participation was very less due to the lack of good communication devices. This was one of the major issues noticed during the PI planning.

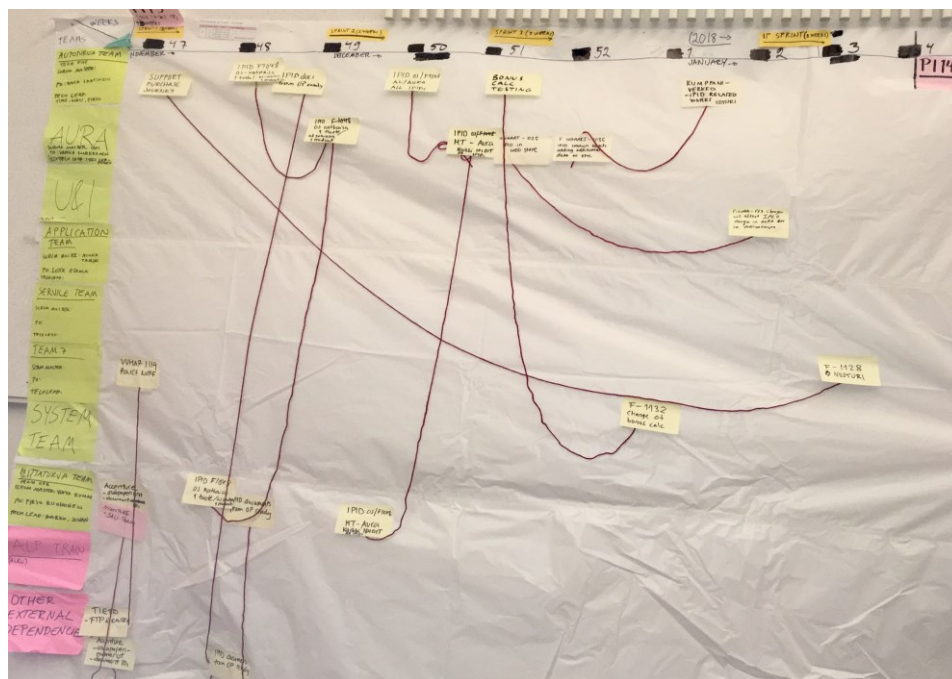


Figure 12: PI Planning - Program board for case company project

Figure 12 is a picture clicked from one of the PI planning meetings, where the product owners or the scrum masters show the feature dependencies for their respective teams on the program board. On the left column, are the team names in a train and the row above is the sprint week. The yellow sticky notes were labelled with the feature number and marked their dependencies on other teams with the red thread. This gives a visual representation of a big picture of how the features were dependent on different teams in a train. This program board enabled to produce transparency and helped in managing the work better. But unfortunately, this level of transparency was minimized for the offshore teams while creating this program board. They only get to view this program board after it was completed and then shared as a picture via email.

The management reviews the program board and makes adjustments based on the challenges and risks, this is again discussed on the second day of the PI Planning meeting. The teams created and presented their final plans and objectives on the second day. During this time, again the business owners such as epic owners circulate across different teams and provide the business value to the team objectives. The program board was updated again based on the changes and the program level risks were discussed in the auditorium when all the team members were present. Finally a confidence vote was taken from all the participants both at onshore and offshore. This confidence vote was based on the scale of number from 1 to 5, where 1 being the lowest and 5 being the highest. This vote of confidence, gives all the members and mainly the clients, a level of confidence on what can be expected. From my analysis of understanding the program board in terms of RE process, it can be considered as part of requirement management process. It could be used for traceability and tracking the dependent feature across all teams in an agile release train (ART).

3. PI execution - Sprint Cycle

After the PI planning, PI objectives were executed using scrum method. The case company project has 4-5 sprint cycles and each sprint was planned for two weeks. For each sprint cycle, the scrum master facilitates the sprint planning for the team. In the sprint planning, the user stories were presented by the product owner and the business analyst. The story points were estimated by the team members for each story. The team members proceed with building and testing the stories according to the test plan and the test plan was assisted and reviewed by the test manager. The Scrum master facilitated the demo at the end of each sprint and also facilitated a retrospective meeting. Retrospective meeting consisted of three questions that had to be answered by each team. The questions were, what went well in the sprint, what didn't go well and what can be improved. This was done by each team and later brought together at the program level during Inspect & Adapt which will be discussed in the coming sections.

It was observed that in some cases there was lack in communication and coordination in the teams across ART. For example, if there were any doubts or queries from the offshore team, they were not very comfortable to directly communicate with the epic owner at the client side but would prefer to communicate via the onshore team. The reasons could be the language or cultural behavior. For each sprint cycle, the sprint demos were given by the team internally and was mainly led by the business analyst or the test engineer. After the sprint demo, the product owner presents the same demo to the client or the epic owner. After the epic owner was satisfied with the sprint demo and the results, the product owner accepted the story implementation and ensured that the user story was completed. The acceptance testing for the user stories and features were done by the acceptance tests specialists at the client side. There was very minimal information gathered on this as it was done at the client location. Table 14 provides a detailed understanding of some important roles and responsibilities that take place in the RE process at team level during PI execution.

Table 14: Roles and responsibilities at team level during PI execution

Level	Role	Location	Responsibilities
Program/Team	Product Owner	Company side (onshore)	<ul style="list-style-type: none"> ➤ Leads the sprint content planning ➤ Accepts/declines the sprint outcome ➤ Co-ordinates with the client and the team both in onshore and offshore ➤ More on the customer facing side ➤ Gives sprint demos to the client
Team	Business Analyst	Company side (offshore)	<ul style="list-style-type: none"> ➤ Co-ordinates with the product owner and the team both in onshore and offshore ➤ More on the team facing side ➤ Gives sprint demos to the product owner
Team	Team Member - (developers and testers)	Company side (onshore & offshore)	<ul style="list-style-type: none"> ➤ Participates in the sprint planning lead by scrum master ➤ Creates, splits and gives relative job size estimates to the user stories with Product Owner ➤ Plans the implementation by splitting the stories into tasks and estimates the time needed to do the tasks ➤ Implements and tests the stories ➤ Participates in the sprint demo and sprint retrospective
Team	Scrum Master	Company side (offshore)	<ul style="list-style-type: none"> ➤ Facilitates the sprint planning and team work ➤ Guides the team in operational work methods and ensures the possibility for development ➤ Ensures that team follows the scrum procedures ➤ Helps to clean up the barriers and impediments from team's work, so that team is able to concentrate on implementing ➤ Escalates when needed to RTE and to relevant functions ➤ Represents team in PI Scrum of Scrums meetings

PI Inspect & Adapt

The scrum teams first internally have their own inspect and adapt meetings after each sprint, they are termed as retrospective meeting. A retrospective meeting helps the teams to recognize improvement points and reports those to Scrum Master at the end of each sprint. These points were collected by the RTE from the each team scrum masters and later brought together at the program level during Inspect & Adapt meeting. The retrospective and Inspect & Adapt meetings were highly beneficial for me to understand the issues not just in my team but also the teams across the train. My area of research got more interesting by especially participating in PI Inspect and Adapt.

4.1.2 Meeting Composition

The Scaled Agile Framework requires some ceremonies to be followed such as meetings. Table 15, summarizes important meetings that have been identified for program and team level participants related to requirements engineering. The meetings at that take place at the portfolio level were mainly organized by the clients and had to be de-scoped in our research study.

Table 15: Meeting composition at program and team level

<p><u>PI planning</u></p> <p>Facilitator - Release train engineer (RTE)</p> <p>Participants - clients (epic owners, portfolio management), architecture, and agile release train consisting of all 8 teams. Members both from onshore and offshore</p> <p>Aim - plan and commit to PI objectives</p> <p>Time - 2 day event</p> <p>Frequency - once in 8-10 weeks</p> <p>Mode of Communication - Face to face in an auditorium and audio conferencing (WebEx or Skype business)</p> <p>SAFe Level - Program</p>	<p><u>Sprint Planning</u></p> <p>Facilitator - Scrum Master</p> <p>Participants - only team members including product owner, scrum master, business analyst, developers and testers. Members both from onshore and offshore</p> <p>Aim - plan what user stories to take up for the sprint and estimate story points</p> <p>Time - 1 hour</p> <p>Frequency - once in every 2 weeks of Pi execution</p> <p>Mode of Communication - audio conferencing (WebEx or Skype business)</p> <p>SAFe Level - Team</p>
<p><u>Scrum of Scrums</u></p> <p>Facilitator - Release Train Engineer (RTE)</p> <p>Participants - SM, PO, Architects, clients (epic owners)</p> <p>Aim - to identify dependencies between the teams, verify risks, update team progress towards PI objectives, product backlog progress</p> <p>Time - 1 hour</p> <p>Frequency - once a week</p> <p>Mode of Communication - audio conferencing (WebEx or Skype business)</p> <p>SAFe Level - Program</p>	<p><u>Scrum - Daily meetings</u></p> <p>Facilitator - Scrum Master (SM)</p> <p>Participants - only team members including product owner, scrum master, business analyst, developers and testers. Members both from onshore and offshore</p> <p>Aim - discuss day to day activities and any impediments</p> <p>Time - 30 minutes</p> <p>Frequency - every day (5 times in a week)</p> <p>Mode of Communication - audio conferencing (WebEx or Skype business)</p> <p>SAFe Level - Team</p>

<p><u>PO Sync</u></p> <p>Facilitator - RTE (RTE) /Product managers</p> <p>Participants - Product owners across the teams in the train, client</p> <p>Aim - to verify overall PI objectives and plan upcoming PI</p> <p>Time - 1 hour</p> <p>Frequency - once a week</p> <p>Mode of Communication - audio conferencing (WebEx or Skype business)</p> <p>SAFe Level – Program</p>	<p><u>Sprint Demo</u></p> <p>Facilitator - Scrum Master (SM)</p> <p>Participants - Product owner, business analyst, epic or feature owner, scrum master and team members (developers and testers)</p> <p>Aim - execute the workable software to validate the outcome of user stories planned for the sprint</p> <p>Time - 1 hour</p> <p>Frequency - end of each sprint (every 2 weeks)</p> <p>Mode of Communication - audio conferencing (WebEx or Skype business)</p> <p>SAFe Level – Team</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.2 Reflected Problems

This section summarizes about the reflected problems identified from my observations made during the case company project analysis. It is also important to mention that, during my analysis, interviews were conducted for different members in the team including RTE, product owner, scrum masters, developers and testers working in different locations. This helped me in understanding their viewpoints depending on their roles and locations. The other factors of my observations also came from retrospective meetings and PI Inspect & Adapt. Table 16 summarizes the challenges identified during the case company project analysis.

Table 16: Challenges in case company project

Category	Challenges
Ambiguous requirements	Due to lack of domain knowledge and system limitations
	During translating requirements due to language constraints
Challenges with globally distributed teams	Ineffective participation in PI planning due to lack of good communication methods/tools
	No proper methods for requirement traceability at program level during PI planning
	Due to communication and collaboration
Challenges in creating features and splitting to user stories	Features did not fit into one PI
	Difficulties in splitting features to user stories

1. Ambiguous requirements

One of the major problem reflected during the RE process was ambiguity in the requirements or not well defined requirements.

- **Due to lack of domain knowledge and system limitations**
- **During translating requirements due to language constraints**

Due to lack of domain knowledge and system limitations

From the observations made while participating in the PI planning meetings, sprint planning meetings and interviews taken, it was revealed that some of the requirements did not have enough details. For example, some epics and features defined by the epic owners in Jira (which is a tool to track and maintain requirements), did not have enough details that could help the developers or the testers understand. The epic owners are from the client side and they assume and expect that all the extra additional information is already known or understood by the company's product owners or business analysts.

In this project the product owner and some team members were relatively new to the domain and the system. From the interviews, the experienced developers and testers who have been working long with these applications mentioned that they knew what should be done, while others admitted that it would be better to have more details since they lacked system and domain knowledge. Due to the lack of domain and system knowledge, the product owner being new to the team faced a lot of issues in breaking down the features to user stories without having enough details. This created a lot of ambiguity and open questions from the team. The product owner took extra time and effort in finding the people who could provide the necessary domain information or ask for guidance to find the necessary documents. Since a lot of time was spent in gathering those enough details, the user stories were constantly updated and the test cases were created late.

Another example is that, the case company project has more than 100 legacy system applications and there were no comments in the code base workflow or information about the functionality for the developers to understand. The system limitations and lack of domain knowledge made it very difficult for the developers to understand and apply the new set of requirements. Though there were a huge number of documentation stored and distributed across the client and the company network, it was not very helpful. It was mentioned in the interviews by almost all team members that it is very difficult to find any information related to the system or the domain because they do not know where to find it. For the offshore team members who work in different locations, it was more difficult to find and have access to the information because of network security constraints.

During translating requirements due to language constraints

Due to the language constraints, it was observed that in some cases there has been misunderstanding in translating the requirements from one language to another. In this case it was from Finnish to English language. Since we work with distributed teams, the translation was required for the members who are located in different locations such as India and Poland.

For example when the epics and features were first created by the epic owner, they were done in Finnish language and eventually translated to English by a translator. The user stories were created in English by the product owner or business analyst, but the issues happened when the features were modified during beginning of sprint and not translated back to English. The project team member such as product owner or a business analyst raises an internal ticket to the translation team to translate the epics and features. This process takes some time and if it was delayed for a longer period, the product owner from Finland itself tries to translate which was an additional responsibility. In some cases the offshore team members used the online translating tools to translate the requirements in English. Such practices can lead to misunderstanding of requirements creating a huge ambiguity amongst the developers and testers. Hence, the ambiguity in requirements also affected the testers where the test cases were constantly updated and some important scenarios were missed for testing the user stories.

2. Challenges with globally distributed teams

- **Ineffective participation in PI planning due to lack of good communication methods/tools**
- **No proper methods for requirement traceability at program level during PI planning**
- **Due to communication and collaboration**

Ineffective participation in PI planning due to lack of good communication methods/tools

SAFe recommends face to face communication during the PI Planning meeting [25]. This is difficult to follow when the team members are located in different locations. Sometimes the company arranges a travel for these members to the main location for planning meetings which has resulted in better understanding and planning of requirements. But this does not happen frequently because of cost constraints. The usual way of communication across team members in different locations for this project is through audio calls. From the interviews taken, some developers and testers located in different locations admitted that they were not very motivated to participate in the requirements planning meeting or PI planning meetings due to the lack of bad audio or video quality. For example, in the PI planning meeting which is held for two days, audio meetings were booked via Skype business or WebEx to communicate to the offshore teams.

During my participation in the PI Planning meetings, I observed that the connections or network were very poor and the offshore team members were unable to follow up the discussions happening at the event. This led to ineffective participation for the offshore team members during PI Planning. Another example is that, when the program board was discussed, where dependencies and risks were shown across teams in a train, they were unable to view or understand what was happening. This was because there were no good video or audio calling tools used. Hence, the team members located in different locations were unable to provide their inputs or contribute their ideas in the PI Planning meeting.

No proper methods for requirement traceability at program level during PI planning.

This is related to the program board where feature dependencies across teams were traced at the program level. As mentioned in the literature study, a program board is a simple physical display that provides a big picture of any feature dependencies across the teams in an agile release train. This board is created during the two day PI planning meeting. The RTE facilitates and takes the responsibility of maintaining the program board. The representation of the board is basically a big white sheet of paper physically pasted to the auditorium wall as shown in the diagram below. The left column includes all the team names in the agile release train (ART) and the rest of the columns are divided by sprint weeks. Sticky notes were used to write the feature number and were pasted to the teams that they are dependent on. The dependencies were connected with a red thread to the sticky notes having the feature number, across the teams. This process was not very helpful for teams located in different location to contribute and understand the dependency between features and relevancy of different teams in a release train. After the PI planning meeting, this physical program board sheet was folded and taken to the project floor and pasted to one of the walls. In this process, some of the sticky notes were lost or misplaced hence creating loss of traceability. These program board sheets were pasted on the walls and replaced every 2-3 months. For a company having multiple trains and multiple program boards, almost all the walls were covered with these sheets.

Due to communication and collaboration. When scaling agile in an enterprise having globally distributed teams, one of the common problems is to align the offshore teams to the same business goal as in house members (onshore). The main issue being communication and collaboration. Collaboration over distance was difficult and during the requirements engineering phase, it was very important that all the team members are aligned on same level of understanding the requirements. It was observed that in some cases, the product owner and the business analyst were located at the onshore location, leaving the offshore location members without any requirement analyst. The members at the offshore location were always dependent on the onshore members for understanding the requirements.

3. Challenges in creating features and splitting to user stories

- **Features did not fit into one PI**
- **Difficulties in splitting features to user stories**

Features did not fit into one PI. From some of the observations made as a business analyst in the team and by conducting interviews for product owners, it was noticed that there were some challenges in creating features and splitting them into user stories. When features were created from an epic, it was created in a way that it fits into one program increment (PI) so that the goal of the business requirement is achieved. If the feature did not fit into one PI, then it got extended till the next PI creating unnecessary anticipation for another 8-12 weeks. This was one of the most common case across the teams in the train, where the features did not fit into one PI. When features are created from an epic at the program level, it is important to analyze and estimate that it fits into one PI.

Difficulties in splitting features to user stories. Some product owners and business analysts found it difficult to split the feature to stories. Sometimes the stories were too big or too small. For example, in one of the sprints, none of the planned user stories were completed due to some technical constraints and dependencies on other teams and they were pushed to the next sprint. The metrics in the tool showed that the result of delivery of user story for that sprint was almost zero. This again continued for the next sprint till the product owner and business analyst decided to split these user stories further to complete the tasks in an agile way. There was a special case in one of the projects where, there were no features created at all and directly have user stories deriving from the epics.

4.3 Good RE Practices for Improvements

To overcome the challenges faced by the case company project, some good RE practices are identified and suggested to improve the RE process in SAFe. Further, this section explains on how the practices were executed in the case project for two consecutive PIs and what was the outcome or results after execution. The results are the output of the work observed after each PI and discussed in PI Inspect & Adapt. In addition to this, there were personal feedbacks taken from the team members both at the onshore and offshore location. The table 17 summarizes the RE practices of SAFe that can be used to overcome the challenges and improve the RE process.

Table 17: Suggested good RE practices of SAFe

Good RE practices	Short description
Applying modelling techniques	Representing information visually by applying modelling techniques such as requirements model and domain model.
Sharing knowledge through Community of Practice (CoP)	This practice enables knowledge sharing by gathering groups of people sharing common interest in technical or business domain.
Maintain requirement traceability for dependencies	This practice acts as a good communication method across globally distributed teams to manage and maintain requirement dependencies across multiple teams.
Work and improve collaboratively	A resource model is proposed in this study to enable communication and collaboration within globally distributed teams to develop shared understanding of RE process.
Learning the RE process of SAFe	Working in SAFe needs training and the framework needs to be well understood to apply and achieve quality results in RE process.

1. Applying ‘modelling’ techniques

The existing and newly joined team members such as developers, testers, product owners and business analyst continuously brought the concern that domain level understanding is a challenge and epics and features do not completely provide a good understanding of the business expectation. The dependencies and system integration complexities created difficulty in scheduling features for a specific iteration and unfinished user stories observed in multiple iteration or sprints. During my discussion with product owners and business analysts, I presented the idea of modelling, mainly focusing on domain and requirement models to overcome the issues. For implementing this practice, the modelling techniques proposed for domain modelling were UML and E-R diagrams and use-case diagrams for representing the requirements.

Execution

For implementing this practice, the tool used to create the diagrams were Microsoft Visio and other online available tools. The intention was to visually represent the information, for example, the visual representation of the requirements enables to learn the system quicker by all parties and overcome the major gaps in complete system integration projects. First we started with representing the features and user stories by creating a use-case diagram at the team level. For the domain model, it took an extra effort in collecting all the required information related to the insurance domain, products and applications. This required a continuous effort.

Results

Along with documenting requirements with epics, features and stories, modelling the requirements improved better understanding in distributed teams breaking the language barriers spoken across. The main feedback from the interviews and PI inspect & adapt meeting was that, it improved communication, quality, productivity and reduced risks. The product managers and product owners also mentioned that it helped them to identify some hidden requirements at an early stage of requirement analysis and it also made it easy for them to verify and validate the requirements. Domain and system modelling highly benefited the members who were new to the team as it helped them to provide some good system and domain knowledge. It reduced their stress of finding updated documents from huge storages, not being sure if they were the latest. In general modelling techniques benefited all team members from different roles such as stakeholders, product owners, business analysts, architects, developers and testers and it was most benefited to communicate with the distributed teams. However, there was a disadvantage where, some stakeholders found it difficult to understand the model and the flow.

2. Sharing knowledge through Community of Practice (CoP)

Community of Practice (CoP) is a SAFe practice of sharing knowledge. As mentioned in the literature study, it can be divided in terms of role such as Scrum masters, product owners or business analyst or RTEs. Each can form a CoP group based on their roles. As part of my research study, this practice was considered as a good opportunity to propose sharing of knowledge on domain, system and role based skills to the groups.

Execution

It was agreed with the SAFe coach of the company that we set a plan on the topics and areas of interest to share knowledge for the role based members. We followed the eight characteristics of successful CoPs proposed by Paasivaara and Lassenius [48]. The eight characteristics were, interesting topic with concrete benefits to participants, passionate leader, proper agenda, decision making authority, open community, supporting tools to create transparency, suitable rhythm, and cross-site participation when needed. In the case project, it was first implemented for all product owners and business analysts across the company. From my interviews and observations, we agreed that we focus on knowledge sharing of the domain, systems or applications in the company and some role based skills such as how to create or write a good user story and etc.

A schedule was arranged to set up CoP meetings once or twice a month. The next step was to identify the experts across the teams who had good knowledge in specific areas who were ready to contribute their time in the CoP meetings to share and gain valuable information. These meetings were mainly facilitated by the SAFe coach. A room was booked for this meeting, which had a sufficient space to accommodate people and also a conferencing call was set up for people who were working from different locations. Transparency was maintained by recording all the important information discussed and shared in the meeting in the organization wiki tool.

Results

The concept of sharing knowledge through this practice attracted members to participate in the CoP meetings. The participating members provided a feedback saying that it was quite effective and helpful to share information. They also mentioned that, it helped in knowing people across the organizations who faced similar or different kinds of issues and how they solved them. Some product owners discussed about their success factors in their projects which benefited the other product owners facing issues, for example, some techniques were discussed on how to split features to user stories and how to plan the features with the epic owners to fit them into one program increment (PI). After seeing the success of these sessions for product owners, this practice was applied for other role based groups such as scrum masters and RTEs.

3. Maintain requirement traceability for dependencies

This practice was proposed for the issues faced on tracing the requirement dependencies in a program board during the PI Planning meeting. A program board is a physical display of requirement feature dependencies across the teams in an agile release train. This was recognized as a failure when the teams are globally distributed and cannot physically participate in the same location. The new approach to trace the feature dependencies was to digitize the program board so that it is easily viewed and tracked for the distributed teams without any loss of information.

Execution

A lot of research and thought process was done on how to visually communicate the program board to the distributed teams working in different locations, in a cost effective way. There were some available online tools such as 'Big Picture' that enables you to replicate the program board digitally and allows to plugin to the Jira tool which is basically used for tracing the requirements. This idea of using the online tool was rejected due to cost constraints. The cheaper and easy solution proposed was to replicate the physical display of the program board in a digitized form via a simple Visio tool. This required a manual effort by someone during the PI Planning meeting. Me as a business analyst did this visualization of program board and it was eventually shared with the distributed teams and uploaded as a PDF format in Jira. This gave the distributed teams a clear sophisticated picture on the dependencies. The figure 13 represents the digitized version of the program board.

Execution

In an Agile release train (ART), the Product Management, Product Owner and Business Analyst take extra responsibility and play a very important role to coordinate and collaborate between globally distributed teams. Figure 14 is a resource model proposed, which helps in understanding how the different roles in the ART can support in coordinating with the offshore teams. The guiding team from the onshore location takes the product and technology ownership. The product management and product owners work in this team in close collaboration with the customer. On the other side, the remote or the outsourced teams (offshore) could be based in a different location and they mainly consists of other team members including developers and testers. It was recommended to have a business analyst at the offshore location who acts like a proxy to the product owner. Similarly, the RTE from the guiding team coordinates with Scrum master in the outsourced team.

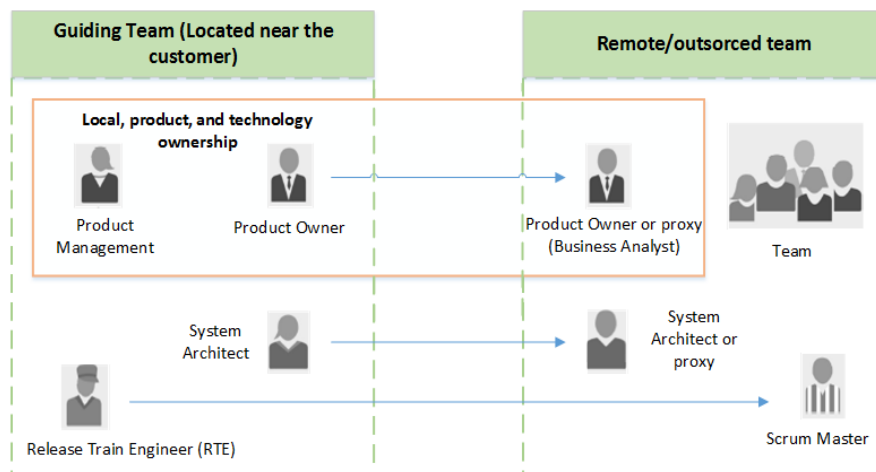


Figure 14: Resource Model

Results

One of the key success factors in any project or organization using agile or scaled agile is communication and collaboration. Hence, the practice of working and improving collaboration helped in achieving good relationship and communication across team members working in different locations. The resource model, helped members connect and collaborate together, sharing responsibilities and information, building a positive team culture.

5. Learning the RE process of SAFe

Working in SAFe needs training and the framework needs to be well understood to apply and achieve quality results in RE process. This practice is already applied in the case company project. Since, it is a Scaled Agile Framework (SAFe) partner, it has certified SAFe coaches.

Execution

The courses are well organized and trained by the coaches on various areas such as SAFe framework 4.5, SAFe for Product Owner/Product Manager, SAFe for Scrum Master, SAFe RTE, SAFe for teams. At the end of the course, it provides certification training to the members and consultation services. This practice is encouraged to all the members in the organization such as business and IT executives, program and project managers, test managers, product owners, developers, testers and etc.

Results

Though the case company has invested in offering trainings and coaching to members to enable valuable results, some members are less motivated to attend the trainings, including the customers. Some members mentioned that it is good in theory but find it difficult to apply their learnings in the real world. However, most of the members from different projects reported that they were benefited a lot from these trainings as it helped them to gain knowledge on the framework and the processes. They also mentioned that it was more valuable to attend the course for 2-3 days rather than reading a thick book. The training materials, consulting services and SAFe expert coaches helped them to solve the issues in teams and ARTs. For example, some product owners in teams struggled to split features into user stories or the teams struggled to fit the feature into one PI. All these challenges were solved by the services provided by the case company project. Figure 15 represents the feedback given by members after the SAFe trainings.

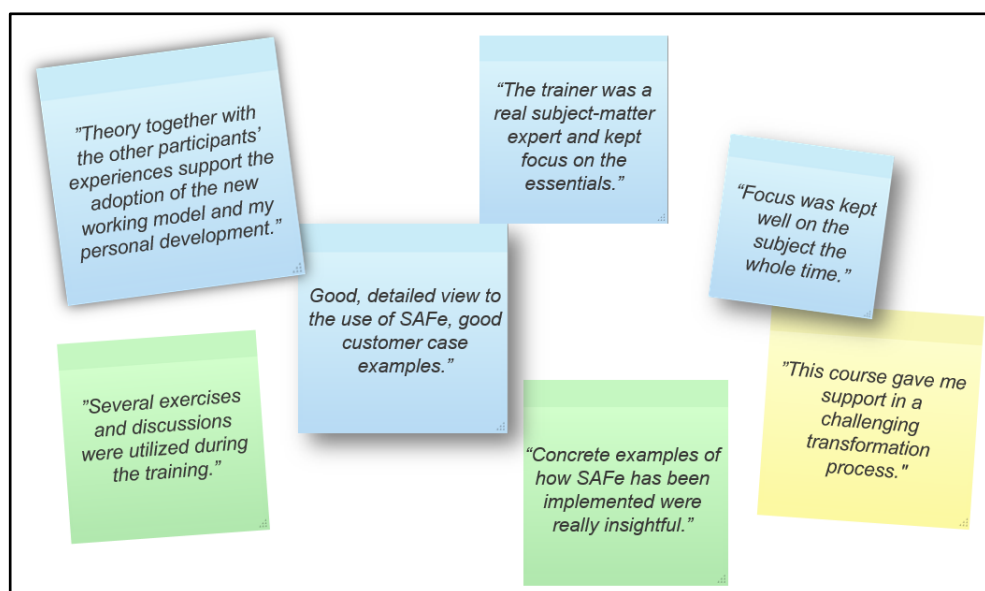


Figure 15: Feedback on SAFe training

4.4 Lessons learned

This section covers the third research question: What are the lessons learned from applying the good RE practices in globally distributed teams. After executing the practices the results are validated in the case company project, by creating a mapping table. Table 18 is a mapping table that, maps the suggested good RE practices against the challenges faced by the case company project.

Table 18: Mapping good RE practices against case company challenges

Challenges in case company project	Suggested good RE practices				
	Applying modelling techniques	Sharing knowledge through Community of Practice (CoP)	Maintain requirement traceability for dependencies	Work and improve collaboratively	Learning the RE process of SAFe
Ambiguous requirements					
Due to lack of domain knowledge and system limitations	x	x		x	
During translating requirements due to language constraints	x				
Challenges with globally distributed teams					
Ineffective participation in PI planning due to lack of good communication methods/tools	x		x	x	
No proper methods for requirement traceability at program level during PI planning			x		x
Due to communication and collaboration	x	x	x	x	
Challenges in creating features and splitting to user stories					
Features did not fit into one PI		x			x
Difficulties in splitting features to user stories		x			x

x - denotes the mapping of the good RE practices with the challenges

From the mapping table 18, it can be observed that, the knowledge sharing through ‘Community of Practice’ helped in overcoming the challenge of ambiguous requirements due to the lack of domain knowledge and system limitations. It also helped in overcoming the challenges in planning requirements such as fitting features into one PI and on how to split features into user stories. This practice also helped in socializing, understanding different work cultures and build good communication and collaboration with globally distributed teams. The practice of learning the RE process of SAFe helped in overcoming the challenges faced during planning of requirements at program and team level. It also helped in having a well-organized PI planning meeting which provided a big picture of requirements to all team members in distributed teams.

Using modelling techniques to represent requirements and domain in the form of use-case diagram or an entity-relationship diagram reduced the ambiguity in requirements. As shown in the mapping table 18, it also helped in overcoming some of the challenges faced with globally distributed, hence improving the RE process. Similarly, the other RE practices such as maintaining requirement traceability and working and improving collaboratively, helped in developing clear and shared understanding of RE process within globally distributed teams.

Another way of analyzing this mapping table is that, different practices together also helped in solving the challenges and improving the RE process in globally distributed teams. The key lessons learned from applying the good RE practices in globally distributed teams are:

Modelling was a good RE practice to be applied in SAFe for visualizing and providing a common understanding of requirements for globally distributed teams.

This RE practice was used to create requirements model and domain models. Modeling the requirements in terms of use-case diagram or entity-relationship diagram provided a visual representation of the system to be built. Along with documenting requirements with epics, features and stories, modelling the requirements provided a common understanding in distributed teams. Modelling techniques benefited all team members from different roles such as stakeholders, product owners, business analysts, architects, developers and testers. The main feedback from different stakeholders was that modelling requirements improved communication, quality, productivity, and reduced risks making it one of the good RE practices. However, the disadvantage was that some business stakeholders found it difficult to understand the requirements model.

Community of Practice (CoP) was a good RE practice of SAFe applied in sharing knowledge and information across different user roles.

SAFe recognizes the general problems in teams and resources such as knowledge gap both in domain and the system/applications. Community of Practice (CoP) was a practice that was used for sharing knowledge among stakeholders and the groups of the different stakeholders were formed based on the roles. The CoP meeting sessions were organized and facilitated by a SAFe coach. The SAFe coach and the experts across the teams having good knowledge in specific areas contributed their time to share valuable information on domain and the system. With the success of the CoP for product owners, this practice was applied for other role based groups such as Scrum masters and RTEs.

It was essential to have SAFe training to develop clear and shared understanding of framework and RE process

Since SAFe is a complex framework, it needs training to be well understood to apply and achieve quality results in RE process. The case company has invested in offering trainings to members which provided valuable results. It enabled the team members to work confidently and be positively inclined towards working with SAFe. It also benefitted in having a well-organized PI Planning meeting. For instance, learning the RE process of SAFe provided good inputs to the Release Train Engineer (RTE) to schedule and plan a well-organized PI planning meeting for globally distributed teams. The result of the PI planning meeting helped to achieve the big picture of requirements to all members of distributed teams.

The practice of ‘work and improve collaboratively’ helped in achieving better coordination and managing of requirements with globally distributed teams

With globally distributed teams, one of the common problems is to align the offshore teams to the same business goal as in house members (onshore). A resource model was proposed which helped the different roles in agile release train (ART) at the onshore location to coordinate with the offshore team. The main roles included Product Management, Product Owner, Business Analyst, RTE and Scrum Master. Since the product and technology ownership is at the onshore location, the product management and product owners worked in close collaboration with the customer. On the other side, the remote or the outsourced teams (offshore) was based in a different location and they mainly consisted of team members such as developers and testers. It was recommended to have a business analyst at the offshore location who acts like a proxy to the product owner. The resource model, helped members connect and collaborate together, sharing responsibilities and information, managing requirements better and building a positive team culture.

Maintaining requirements traceability for dependencies across globally distributed teams, helped in creating transparency

This practice was proposed for the issues faced on tracing the requirement dependencies in an effective way. In the PI planning meeting, a program board is created which is physical display that provides a big picture of any feature dependencies across the teams in ART. But this was recognized as a failure for globally distributed teams as they could not physically participate in the same location to view or provide their inputs. The digital representation of program board in the PI planning meeting, acted as a good communication method in addition to a good audio/video setup. The members from the offshore location India mentioned that the digitized program board helped them gain clear visibility and understanding of feature dependencies. However, this required some manual effort by someone in the team to create the paper program board to the digitized form.

The PI planning meeting helped members of globally distributed teams understand better the big picture of requirements

PI planning meeting is good way of conveying information to the team members in a face-to-face conversation. Different stakeholders attend this event personally if possible or connect remotely. However, in globally distributed teams, the event occurs at multiple locations simultaneously, with real-time communication between the locations. To make this event successful for globally distributed teams, it was important that it was organized well by the Release train engineer (RTE) of ART. It was also important that all the different stakeholders in ART follow the practice of learning the RE process of SAFe. This good RE practice of SAFe helped in understanding the framework well to apply and achieve quality results in PI Planning meeting. For example, it provided good inputs to the Release train engineer (RTE) to schedule and plan a well-organized PI planning meeting for globally distributed teams. The result of the PI planning meeting helped to achieve the big picture of requirements, and to make a commitment to an agreed set of objectives for the next PI.

5. Discussions

5.1 RQ1: Current RE process using SAFe in the case company

The first research question that was answered during this thesis is '*what is the current state of RE process using Scaled Agile Framework (SAFe) in the case company*'? The analysis of RE process in SAFe in literature study was taken as a basis in empirical study to analyze the current RE process in the case company project. The goal was to understand the RE activities, roles and responsibilities at the different levels of SAFe and identify the challenges.

Current RE process of case company project using SAFe

RE process in SAFe starts at the portfolio level and continues to the program and team level. SAFe describes well defined roles at each level in the framework. Each role has certain responsibilities based on portfolio, program or team level. Since the research study was based on the RE process, my focus was to concentrate only on the events that connected with RE process in SAFe. Therefore, identifying the RE activities, user roles and responsibilities at each SAFe level helped in analyzing the current RE process in case company project using SAFe.

Portfolio level activities focus on discovering new ideas which can also be called as business requirements [26]. The lean portfolio management and epic owners work together on eliciting the new ideas. The epic owner writes the epics and later splits them into features. In the current case company project, all the portfolio level activities took place at the onshore location mainly at the customer location.

Program level activities included splitting of epics into features by the epic owner and product management. Once the features were ready, they were prioritized and allocated to different teams in an Agile Release Train (ART) at the program level. The prioritization of the features were changed during the PI planning when all the members participate and identify dependencies and risks. In the current case company project, all the program level activities took place at onshore location.

Program Increment (PI) is an 8-12 week iterative process of agile software development life cycle, starting from the RE process, development, testing and deployment [26]. Program Increment is preceded by a PI Planning meeting which is a face-to face event including different stakeholders from all levels [26]. The PI planning meeting was driven by the RTE and it took place mainly at the onshore location where the clients were also present. For offshore located teams, the event occurred simultaneously with real-time communication by using audio setup and internal meeting links such as Skype and WebEx. The PI planning meeting in this case company project was held for 2 days. The day one started with the business owners presenting their business context, product vision or roadmap.

During the planning session, a list of epics and features were discussed at a higher level. This approach of sharing the business context and vision provided a good understanding of requirements to all the members in the project including the technical team. The output of the PI Planning meeting is a Program board and PI committed objectives. Program board is a physical display that represents feature dependencies across teams in ART [26]. The program board in terms of RE process could be related to requirement management process. It could be used for traceability and tracking the dependent feature at a high level across all teams in ART.

Team level activities included splitting of features into user stories by the product owner and business analyst. During the PI planning meeting, each team in ART analyzed the features and stories allocated to them. Each team identified risks and feature dependencies and discussed it with other teams. The feature dependencies across teams in ART were represented on the program board. It was observed that during the discussion of program board, the offshore team's participation was very less due to the lack of good communication devices. In the current case company project, the team level activities took place both at the onshore location and offshore location.

Reflected problems in RE process of case company project using SAFe

As an active participant in the case company project, some observations were made while analyzing the RE process in SAFe. Also interviews were conducted for different members in the team including RTE, product owner, scrum masters, developers and testers working in different locations. This helped me in understanding their viewpoints depending on their roles and locations. All these factors helped in identifying the problems of RE process in case company project using SAFe.

1. Ambiguity in the requirements

- **Due to lack of Domain Knowledge and system limitations:** The newly joined team members such as developers, testers, product owners and business analyst continuously brought the concern that understanding the domain was a challenge. Also, the case company project uses legacy system applications which limited the information about the functionality for the developers. The system limitations and lack of domain knowledge made it very difficult for the developers to understand and apply the new set of requirements.
- **During translating requirements due to language constraints:** Due to the language constraints, there were misunderstanding in translating the requirements from one language to another. In this case it was from Finnish language to English language. Since the case company project works with globally distributed teams, the translation was required for the members who were located in different locations such as India and Poland.

2. Challenges with globally distributed teams

- ***Ineffective participation in PI planning due to lack of good communication methods/tools:*** During the PI Planning meetings it was observed that the offshore team members were unable to follow up the discussions happening at the event due to weak network and audio connections.
- ***No proper methods for Requirement traceability at program level during PI planning:*** This is related to the program board which is a physical display that represents feature dependencies across teams in ART. This process was not very helpful for teams located in different location to contribute and understand the dependency between features.
- ***Due to communication and collaboration:*** communicating and collaborating over distance was difficult with distributed teams during RE process. The product owner and the business analyst were located at the onshore location, leaving the offshore location members without any requirement analyst. The members at the offshore location such as developers and testers were always dependent on the onshore members for understanding the requirements.

3. Challenges in creating features and splitting to user stories

- ***Features did not fit into one PI:*** When epics were split into features at the program level, it was important to analyze and estimate in a way that it fits into one program increment. The product management found it difficult to plan the features, hence extending it to next PI creating unnecessary anticipation for another 8-12 weeks.
- ***Difficulties in splitting features to user stories:*** The product owners and business analyst faced challenges in splitting features to user stories. They were either big or too small. The teams were unable to complete the big stories in one sprint.

5.2 RQ2: Good RE practices applied in RE process using SAFe

The second research question that was answered during this thesis is '*which good RE practices can be applied in RE process using Scaled Agile Framework, of the case company?*' The goal of this research question was to identify and implement the good RE practices that can be applied in RE process of SAFe. The results of the good RE practices helped in overcoming the challenges faced in the case company project, using SAFe.

- ***Applying modelling techniques:*** This good RE practice represents information visually by creating models such as, requirements model and domain models. For implementing this practice, the modelling techniques proposed for representing domain and requirements were entity-relationship diagrams and use-case diagrams. Along with documenting requirements (epics, features and stories), modelling improved better understanding of requirements in distributed teams.

The visual representation reduced ambiguity in requirements and made it easy to communicate by breaking the language barriers spoken across. Entity-Relationship diagrams used for modelling the domain information, benefited the members who were new to the team as it helped them to provide good domain knowledge. The tools used to create the models were Microsoft Visio and other online available tools.

- ***Sharing knowledge through Community of Practice (CoP):*** Community of Practice enables knowledge sharing by gathering groups of people having common interest in technical or business domain [25]. SAFe uses this concept as a practice to enable practitioners to exchange knowledge and skills with people across entire organization [25]. The groups can be divided into roles such as scrum masters, product owners or business analyst or RTEs. In the empirical study the practice was implemented for the product owners and business analysts to share their skills and experiences to improve the RE process. We followed the eight characteristics of successful CoP proposed by Paasivaara and Lassenius [48]. The eight characteristics were, interesting topic with concrete benefits to participants, passionate leader, proper agenda, decision making authority, open community, supporting tools to create transparency, suitable rhythm, and cross-site participation when needed. Community of Practice helped the members in knowing people across the organizations who faced similar or different kinds of issues and how they solved them. Some product owners discussed about their success factors in their projects which benefited the other product owners facing issues. For example, some techniques were discussed on how to split features to user stories.

- ***Maintain requirement traceability for dependencies:*** This good RE practice of SAFe manages and maintains requirement dependencies. In the PI planning meeting, a program board is created which is physical display that provides a picture of feature dependencies across the teams in an Agile Release Train (ART). SAFe believes that, creating a program board enables face-to-face communication with team members on discussing the feature dependencies. But this was recognized as a failure when the teams were globally distributed and cannot physically participate in the same location. It was proposed to digitize the program board so that it could be easily viewed by the globally distributed teams without any loss of information. The cheaper and easy solution proposed to replicate the physical display of the program board to a digitized form, was by using a Microsoft Visio tool. This resulted in solving most of the challenges faced with the distributed teams. This gave the offshore teams a clear visibility and understanding of feature dependencies, and helped them participate effectively in the PI Planning meetings. However, this required some manual effort by someone in the team to create the paper program board to the digitized form
- ***Work and improve collaboratively:*** The purpose of applying this good RE practice of SAFe was to enable good communication and collaboration within globally distributed teams to develop shared understanding of RE process in SAFe. With globally distributed teams, one of the common problems is to align the offshore teams to the same business goal as in house members (onshore). As part of this practice, a resource model was proposed which helped the different roles in agile release train (ART) at the onshore location to coordinate with the members in the offshore team. It was recommended to have a business analyst at the offshore location who acts like a proxy to the product owner from onshore location. The business analysts helped in communicating the requirements to the offshore teams. The resource model helped the members to work and collaborate together, by sharing responsibilities, information, and build a positive team culture.
- ***Learning the RE process of SAFe:*** Since SAFe is a complex framework to understand, it requires training to have a better understanding of the process and achieve quality results in RE process. The teams without adequate training and coaching struggled with applying agile practices correctly in the RE process [4]. The case company project provided a well-established training and coaching system to enable people to learn the RE process of SAFe. This good RE practice applied in SAFe helped in overcoming the challenges faced during planning the requirements such as creating features that fits into one program increment (PI) and splitting them into user stories. It also helped in having a well-organized PI Planning meeting which provided a big picture of requirements to all members in distributed teams.

5.3 RQ3: Lessons learned from applying good RE practices in globally distributed teams

The third research question that was answered during this thesis is ‘*what are the lessons learned from applying good RE practices in globally distributed teams using SAFe?*’ The goal of this research question is to identify the key lessons learned, from applying good RE practices in globally distributed teams, to improve the RE process in SAFe.

Applying modelling techniques was a good RE practice for visualizing and providing a common understanding of requirements for globally distributed teams using SAFe.

This good RE practice enabled to represent requirements visually by creating a requirements model such as use-case diagram. The visual representation reduced ambiguity in requirements within the globally distributed teams. Hoffmann and Lehner, also proposed modelling and prototyping to be used as a good RE practice, to eliminate requirement specification ambiguities and inconsistencies [6]. Along with documenting requirements with epics, features and stories, modelling the requirements provided a common understanding in distributed teams. Modelling requirements improved communication, quality, productivity, and reduced risks making it one of the good RE practices.

Community of Practice (CoP) was a good RE practice of SAFe applied in sharing knowledge and information in different user roles to improve RE process.

Community of Practice (CoP) enabled knowledge sharing by creating group of experts who shared a common interest or topic and collectively want to deepen their knowledge. This good RE practice of SAFe helped in overcoming the challenge of ambiguity in requirements due to knowledge gap in domain and system/applications. The eight characteristics of a successful CoP suggested by Paasivaara and Lassenius [48] was applied in the empirical study of this thesis. The eight characteristics are: interesting topic with concrete benefits to participants, passionate leader, proper agenda, decision making authority, open community, supporting tools to create transparency, suitable rhythm, and cross-site participation when needed [48].

Since this research study was focused on the RE process of SAFe, CoP meetings were first organized for product owners and business analyst. The topics were related to the business domain, system knowledge, skills and techniques that were required in RE process. The CoP meetings were organized and facilitated by the SAFe coach. The experts were identified based on the topic and the meetings had a good agenda. A room was booked for this meeting, which had a sufficient space to accommodate people and also a conferencing call was set up for people who were working from different locations. Transparency was maintained by recording and publishing the important information in the organization wiki tool. With the success of the CoP for product owners, this practice was applied for other role based groups such as Scrum masters and RTEs.

It was essential to have SAFe training to develop clear and shared understanding of framework and RE process

Learning the RE process of SAFe was important to achieve quality results in RE process. SAFe framework could be difficult to understand as it has a set of roles and RE activities at each level of SAFe. The portfolio level of SAFe was responsible to create epics. The program level activities included splitting of epics into features and organizing PI planning meeting. The team level activities included, splitting of features into user stories and implementing them. One of the guidelines for RE process improvement is to ‘train all process users’ [4]. As mentioned in the literature study, the good RE practice of training members was considered as one of the success factors that supported organizations for wide implementation of RE processes [4]. The case company has invested in offering trainings and coaching to members which provided valuable results. It enabled the team members to work confidently and be positively inclined towards working with SAFe. Therefore, it is essential to have the SAFe training to have a good understanding of this RE process flow. It also benefitted in having a well-organized PI Planning meeting. For instance, learning the RE process of SAFe provided good inputs to the Release train engineer (RTE) to schedule and plan a well-organized PI planning meeting for globally distributed teams. The result of the PI planning meeting helped to achieve the big picture of requirements, and to make a commitment to an agreed set of objectives for the next PI.

The practice of ‘work and improve collaboratively’ helped in achieving better coordination and managing of requirements with globally distributed teams

Large organizations that have globally distributed teams should be managed with proper communication and the necessary networking and tooling architecture [45]. A resource model was proposed in this research study which helped the different roles in agile release train (ART) at the onshore location to coordinate with the offshore team. It was recommended to have a business analyst at the offshore location who acts like a proxy to the product owner at the onshore location. This helped in aligning the offshore teams to the same business goal as in house members (onshore). The resource model also helped the members connect and collaborate together, sharing responsibilities and information, managing requirements better and building a positive team culture.

5.4 Limitations of the study

Literature study

It was difficult to find scientific publications related to SAFe, and no case studies on SAFe were found. Therefore, the analysis of RE process in SAFe and its good RE practices were based on the SAFe books. There might be scope for further research to identify good RE practices that have not been identified in this research.

Empirical study

During the current state analysis of the case company project, the requirements gathering and management at the portfolio level remained invisible in this research study. Since, the requirements gathering takes place at the client office location with their stakeholders, it created a limitation in suggesting good practices to improve the RE process at the portfolio level. However, the RE process at program and team level was clearly visible and attainable.

Another limitation was the inconsistency of good RE practices of SAFe followed throughout programs and teams in the organization. A summary of common issues were mainly collected from the teams in the case company project. This was done to get a common idea of issues only in the case company project and not for the whole organization.

Other limitation arises from the fact that the case company project works in globally distributed environment, making it challenging for my action research methodology. It was not possible to observe good RE practices of teams that were working in offshore locations. However, a constant communication through interviews and feedbacks on phone or meeting calls helped in partially achieving the observational goal.

6. Conclusions

The research problem this thesis attempted to answer was stated as follows: *How can the Requirements Engineering (RE) process using Scaled Agile Framework (SAFe) be improved in globally distributed teams?* It sets out to answer this research problem by diagnosing the current state of the case company project using SAFe. The study attempts to identify and address the challenges faced by the case company project by suggesting some good RE practices and then collect the lessons learned. The results indicate that, combining different good RE practices and working collaboratively with the globally distributed teams, helps in clear and common understanding of the requirements in SAFe. The following three statements are the main conclusions of this thesis that answer the research problem.

A well-organized Program Increment (PI) planning meeting is an important RE practice of SAFe in providing the Big Picture of requirements to all members of distributed teams.

PI Planning is an integral part of Scaled Agile Framework as it provides the big picture of requirements to globally distributed teams in large organizations. It is defined as a good method of conveying information to and within a team with face-to-face conversation. Each Agile Release Train (ART) of SAFe has its own PI planning and includes stakeholders from all levels (portfolio, program and team). All stakeholders, members who are involved in the train attend this event personally if possible or connect remotely. However, in globally distributed teams, the event gets a bit challenging and occurs at multiple locations simultaneously, with real-time communication between the locations.

To have a well-organized PI planning, it is important to understand how SAFe supports RE process. The framework needs to be well understood to apply and achieve quality results in RE process. For instance, PI planning requires a good preparation, coordination and communication. Prior to the PI planning meeting, the product management collaborates with the customer, other stakeholders and product owner to develop the roadmap, program vision, and prepare and estimate the epics, features and milestones. Providing a digital representation of program board in the PI planning for tracing requirement dependencies across agile teams in ART, helped in solving most of the challenges faced with the distributed teams. It acted as a good communication method in addition to a good audio/video setup. Therefore, in addition to the requirement traceability practice, the practice of learning the RE process of SAFe helped in planning the requirements at the program and team level. It also helped in having a well-organized PI planning meeting providing a big picture of the requirements.

Community of Practice (CoP) can be a key RE practice of SAFe in sharing knowledge about business domain, system, skills, techniques, and experiences.

SAFe recognizes the general problems in globally distributed teams such as ambiguity in requirements due to knowledge gap both in domain and the system/applications. Community of Practice (CoP) is a SAFe practice that enables knowledge sharing by gathering groups of people sharing common interest in technical or business domain. The groups are formed based on roles such as, scrum masters, product owners or business analyst and RTEs.

The concept of sharing knowledge through this practice attracted more members to participate in the CoP meeting sessions. The groups were encouraged to collaborate regularly to share information, improve their skills and actively work on advancing their knowledge on the domain. The CoP meeting sessions were organized and facilitated by the SAFe coach. The coach and the experts across the teams having good knowledge in specific areas contributed their time to share valuable information on domain and the system. The participating members provided a feedback saying that it was quite effective and helpful to share information. They also mentioned that, it helped in knowing people across the organizations who faced similar or different kinds of issues and how they solved them. Some product owners discussed about their success factors in their projects which benefited the other product owners facing issues. For example, some techniques were discussed on how to split features to user stories and how to plan the features with the epic owners to fit them into one program increment (PI). It was also viewed as a social learning network where people shared their experiences, knowledge and skills to improve the RE process.

Working and improving collaboratively within globally distributed teams is essential for clear and shared understanding of requirements.

With globally distributed teams, one of the common problems is to align the offshore teams to the same business goal as in house members (onshore). The main issue being communication and collaboration. Collaboration over distance is difficult and during the requirements engineering phase, it is very important that all the team members are aligned on same level of understanding the requirements.

One of the approaches was to apply a practice of improving collaboratively by having well defined roles and responsibilities across distributed teams. A resource model was proposed in this study which shows different roles in SAFe, coordinating with offshore teams during RE process. The main roles include RTE, scrum master, architects, product owner and business analyst. The resource model helped team members collaborating together, sharing responsibilities and information, and building a positive team culture.

Future research

The momentum around scaling agile is growing rapidly in large organization and it faced a tremendous growth since 2014. Scaled Agile Framework (SAFe) made a significant jump to become the most popular scaling agile method in 2017 [20]. However, there have been very few scientific publications on SAFe and no case studies on SAFe were found.

In the future, it would be interesting to investigate more on the Scaled Agile Framework and provide case studies on the best RE practices used in Scaled Agile Framework. What are the benefits and success factors of large organizations using SAFe? How can SAFe be used in software development lifecycle of a product? What is the adoption level of SAFe in large organizations?

References

- [1] R. R. Young, “The Requirements Engineering Handbook”. Artech House, 2004.
- [2] B. Nuseibeh and S. Easterbrook, “Requirements Engineering: A Roadmap,” *Proceedings of the Conference on The Future of Software Engineering*, pp. 35–46, 2000.
- [3] F. Paetsch, A. Eberlein, and F. Maurer, “Requirements engineering and agile software development,” in *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003*, pp. 308–313.
- [4] M. Kauppinen, M. Vartiainen, J. Kontio, S. Kujala, and R. Sulonen, “Implementing requirements engineering processes throughout organizations: success factors and challenges,” *Information and Software Technology* 46(14) , pp. 937–953, 2004.
- [5] S. Lauesen, “Software Requirements : Styles and Techniques”, *Addison-Wesley*, 2002.
- [6] H. F. Hofmann and F. Lehner, “Requirements engineering as a success factor in software projects,” *IEEE Software* 18(4), pp. 58-66, 2001.
- [7] I. Sommerville, *Software Engineering, 9th edition*, Nine. New York : Addison-Wesley, 2010.
- [8] D. Pandey, U. Suman, and A. K. Ramani, “An Effective Requirement Engineering Process Model for Software Development and Requirements Management,” in *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, , pp. 287–291, 2010.
- [9] L. Cao and B. Ramesh, “Agile Requirements Engineering Practices: An Empirical Study,” *IEEE Software* 25(1), pp. 60–67, 2008.
- [10] M. Engin, “Research Diary: A Tool for Scaffolding,” *International Journal of Qualitative. Methods* 10(3), pp. 296–306, 2011.
- [11] K. Dikert, M. Paasivaara, and C. Lassenius, “Challenges and success factors for large-scale agile transformations: A systematic literature review,” *Journal of Systems and Software*, 119, pp. 87–108, 2016.
- [12] A. Sillitti and G. Succi, “Requirements Engineering for Agile Methods,” in *Engineering and Managing Software Requirements*, Berlin/Heidelberg: Springer-Verlag, pp. 309–326, 2005.
- [13] I. Sommerville and P. Sawyer, “Requirements Engineering : A Good Practice Guide”, *1st edition*, Wiley, 1999.
- [14] P. Zave, “Classification of Research Efforts in Requirements Engineering,” *ACM Computing Surveys (CSUR)* 29(4), pp. 315–321, 1997.

- [15] A. De Lucia and A. Qusef, "Requirements Engineering in Agile Software Development," *Journal of Emerging Technologies in Web Intelligence* 2(3), 2010.
- [16] J. M. Bhat, M. Gupta, and S. N. Murthy, "Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing," *IEEE Software* 23(5), pp. 38–44, 2006.
- [17] G. Kotonya and I. Sommerville, "Requirements Engineering: Processes and Techniques", *1st edition*, Wiley Publishing, 1998.
- [18] A. G. Sutcliffe, "The Encyclopedia of Human-Computer Interaction, 2nd Ed. 13. Requirements engineering," *Interaction design foundation*, 2009. [Online]. Available: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/requirements-engineering>. [Accessed: 25-Jan-2017].
- [19] Version One, "10th Annual State of Agile Report," 2016.
- [20] Version One, "11th Annual State of Agile Report," 2017.
- [21] B. Boehm and R. Turner, "Management Challenges to Implementing Agile Processes in Traditional Development Organizations," *IEEE Software* 22(5), pp. 30–39, 2005.
- [22] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology* 50(9–10), pp. 833–859, 2008.
- [23] M. Ylikangas and M. Kauppinen, "Domain modelling -Software design and modelling." 2016
- [24] D. Mishra and A. Mishra, "Complex software project development: agile methods adoption," *Journal of Software Maintenance and Evolution: Research and Practice* 23(8), pp. 549–564, 2011.
- [25] SCALED AGILE Inc, "Scaled Agile Framework – SAFe for Lean Enterprises," 2018. [Online]. Available: <https://www.scaledagileframework.com/>. [Accessed: 10-Mar-2018].
- [26] D. Leffingwell, *SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean for Lean Software and Systems Engineering*. Addison-Wesley Professional, 2016.
- [27] D. Leffingwell, *Agile software requirements : lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.
- [28] C. Larman and B. Vodde, "Practices for Scaling Lean & Agile Development", *1st edition*, Addison-Wesley Professional, 2010.
- [29] R. Brenner and S. Wunder, "Scaled Agile Framework: Presentation and real world example," in *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 1–2, 2015.
- [30] M. Fowler and J. Highsmith, "The Agile Manifesto" *Software Development* 9(8), pp. 28–35, 2001.

- [31] Scaled Agile Inc, “Case Study – Scaled Agile Framework,” 2016. [Online]. Available: <https://www.scaledagileframework.com/case-studies/>. [Accessed: 17-Jan-2018].
- [32] M. Paasivaara and C. Lassenius, “Scaling Scrum in a Large Globally Distributed Organization: A Case Study,” in *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, pp. 74–83, 2016.
- [33] Scaled Agile Inc, “Release on Demand – Scaled Agile Framework.” [Online]. Available: <https://www.scaledagileframework.com/release-on-demand/>. [Accessed: 23-Feb-2018].
- [34] “SAFe ® 4.5 Introduction Overview of the Scaled Agile Framework ® for Lean Enterprises,” *A Scaled Agile Inc. White Paper*, 2017.
- [35] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, “Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study,” in *2014 IEEE 9th International Conference on Global Software Engineering*, pp. 16–25, 2014.
- [36] C. J. Stettina and J. Hörz, “Agile portfolio management: An empirical perspective on the practice in use,” *International Journal of Project Management* 33(1), pp. 140–152, 2015.
- [37] J. A. Lara, D. Lizcano, M. A. Martínez, J. Pazos, and T. Riera, “A UML profile for the conceptual modelling of structurally complex data: Easing human effort in the KDD process,” *Information and Software Technology*, 56(3), pp. 335–351, 2014.
- [38] R. L. Baskerville and A. T. Wood-Harper, “A Critical Perspective on Action Research as a Method for Information Systems Research,” in *Enacting Research Methods in Information Systems: Volume 2*, Cham: Springer International Publishing, 2016, pp. 169–190.
- [39] E. Wenger, “Communities of Practice and Social Learning Systems: the Career of a Concept,” in *Social Learning Systems and Communities of Practice*, London: Springer London, 2010, pp. 179–198.
- [40] Scaled Agile Inc, “Epic – Scaled Agile Framework,” 2016. [Online]. Available: <https://www.scaledagileframework.com/epic>. [Accessed: 15-Mar-2018].
- [41] R. Roth, “Write a Great User Story,” *CA Agile Central Help*. [Online]. Available: <https://help.rallydev.com/writing-great-user-story>. [Accessed: 2-Apr-2018].
- [42] T. Dingsøyr and C. Lassenius, “Emerging themes in agile software development: Introduction to the special section on continuous value delivery,” *Information and Software Technology*, 77, pp. 56–60, 2016.
- [43] P. Saddington, “Scaling Agile Product Ownership through Team Alignment and Optimization: A Story of Epic Proportions,” in *2012 Agile Conference*, pp. 123–130, 2012.
- [44] D. Leffingwell and J. Aalto, “A Lean and Scalable Requirements Information Model for the Agile Enterprise,” *White Paper*, Leffingwell LLC, 2009.

- [45] D. Leffingwell, *Scaling software agility : best practices for large enterprises*. Addison-Wesley, 2007.
- [46] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen, “Action research,” *Communications of the ACM*, 42(1), pp. 94–97, 1999.
- [47] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 68–77, 2008.
- [48] M. Paasivaara and C. Lassenius, “Communities of practice in a large distributed agile software development organization – Case Ericsson,” *Information and Software Technology* 56(12), pp. 1556–1577, 2014.
- [49] E. Wenger, *Communities of practice : learning, meaning, and identity*. Cambridge University Press, 1998.
- [50] M. Ylikangas and M. Kauppinen, “Requirements Engineering –From the Perspective of Customers and Users.” *Software design and modelling*, 2016.

Appendix

This section includes the interview questions that were conducted during the case company analysis of RE process in SAFe. The interview questions were created for different roles such as Release train engineer (RTE), product owners, Scrum masters, developers, testers and SAFe coach.

Interview questions for Release train Engineer (RTE)

Background

- What is your background?
- What are your responsibilities and tasks?
- Which location do you work from (onshore or offshore)?
- How well do you understand the Scaled Agile Framework and the RE process?
- How strictly do you feel that the project follows the SAFe Framework?

RE process in SAFe

- What are the other roles in SAFe you interact with during RE process?
- How do you coordinate with multiple agile teams in ART which are globally distributed?
- How do you work to create effective teams?
- What methods are used to communicate with the globally distributed teams during PI Planning meeting?
- What challenges do you face while organizing the PI Planning meeting?
- How do you manage the requirement dependencies across multiple agile teams in agile Release Train (ART)?
- How do you track the Program Increment feature completion?

Interview questions for Product Owners

Background

- What is your background?
- What are your responsibilities and tasks?
- Which location do you work from (onshore or offshore)?
- How well do you understand the Scaled Agile Framework and the RE process?
- How strictly do you feel that the project follows the SAFe Framework?

RE process in SAFe

- Describe briefly how the RE process using SAFe was conducted in the project?
- What are your responsibilities at the portfolio level, program level and team level during the RE process?
- How do you synchronize and manage requirements with the globally distributed team?
- How do you handle dependencies?
- How do you break down your epics to features, features to stories and assign them to the right team?

- How do you prioritize requirements and estimate backlogs?
- How do you manage any change in the requirement?
- What are your responsibilities during PI Planning meeting?
- What methods do you use to interact with your globally distributed teams?
- What challenges do you face during the RE process with the globally distributed teams?
- What challenges do you face while coordinating with the client and team members both at the onshore and offshore?

Interview questions for Scrum Masters

Background

- What is your background?
- What are your responsibilities and tasks?
- Which location do you work from (onshore or offshore)?
- How well do you understand the Scaled Agile Framework and the RE process?
- How strictly do you feel that the project follows the SAFe Framework?

RE process in SAFe

- What are the other roles in SAFe you interact with during RE process?
- What method of communication is used to during meetings such as PI Planning meetings, sprint planning and Scrum
- How do you manage the impediments in the agile team?
- What challenges do you face during the RE process with the globally distributed teams?
- What challenges do you face while coordinating with the team members both at the onshore and offshore?

Interview questions for Developers and Testers

Background

- What is your background?
- What are your responsibilities and tasks?
- Which location do you work from (onshore or offshore)?
- How well do you understand the Scaled Agile Framework and the RE process?
- How strictly do you feel that the project follows the SAFe Framework?

RE process in SAFe

- From your point of view, what is the RE process in SAFe? Could you describe the process in brief?
- At what level in SAFe do you start participating in the RE process?
- What do you think a "good" requirement should be?
- Do you have enough information to understand and implement the requirement?
- Have you misunderstood any requirement? At what circumstances have you misunderstood the requirement?
- What additional information or improvements would you require to understand the requirements better?

- At what stage are the test cases created?
- Do you have enough information in the user story and acceptance criteria, to create the test cases?
- How often do you interact and communicate with the product owner or business analysts?
- What are the other roles in SAFe you interact with during RE process?
- What are your responsibilities during the PI Planning meeting?
- What are the benefits and challenges of attending the PI Planning meeting?
- What challenges do you face working with distributed teams during the RE process
- What challenges do you face while coordinating with the team members both at the onshore and offshore?

Interview questions for SAFe coach

Background

- What is your background?
- Have you worked in any other SAFe role before becoming a SAFe coach? If yes, what was your role and responsibilities?
- What are your responsibilities as a SAFe coach?
- Which location do you work from (onshore or offshore)?
- Do you have SAFe coaches both at the onshore and offshore location?
- How strictly do you feel that the organization follows the SAFe Framework?

RE process in SAFe

- What kind of SAFe trainings do you give? Does the training cover the RE process?
- How are the SAFe training organized?
- How often do you schedule the training?
- How long are the trainings?
- What roles and members in the organization do you train?
- How important is it for the members to learn the RE process of SAFe?
- What level of support do you give to the projects that work with SAFe in the organization?
- Do you have any responsibilities in the project using SAFe during the RE process? If yes, what are your responsibilities?
- What challenges are faced by the members in the projects using SAFe by RE process?
- How does the globally distributed teams impact the RE process in the projects using SAFe?
- What good RE practices of SAFe can you suggest in the project to improve the RE process?